pygeonlp

リリース *1.2.2*

sagara@info-proto.com <Takeshi Sagara>

2023 年 12 月 11 日

目次

Ver. 1.2.2

PyGeoNLP は、普通の日本語テキスト (自然文) を解析し、地名部分を抽出する geotagger や geoparser と呼ばれるツールです。

次の例のように、文中の地名 (「目黒駅」「品川区」) を**地名語** として認識し、それぞれのクラス (「鉄道施設/鉄道駅」「市区町村」) や経緯度などを付与することができます。

```
% echo "目黒駅は品川区にあります。" | pygeonlp geoparse
目黒駅　名詞,固有名詞,地名語,Xy26iV:目黒駅,*,*,目黒駅,,鉄道施設/鉄道駅,Xy26iV,目黒駅,139.
↪71566,35.632485
は　助詞,係助詞,*,*,*,*,は,ハ,ワ
品川区　名詞,固有名詞,地名語,kEAYBl:品川区,*,*,品川区,,市区町村,kEAYBl,品川区,139.73025000,
↪35.60906600
に　助詞,格助詞,一般,*,*,*,に,ニ,ニ
あり　動詞,自立,*,*,連用形,五段・ラ行,ある,アリ,アリ
ます　助動詞,*,*,*,基本形,特殊・マス,ます,マス,マス
。　記号,句点,*,*,*,*,。,。,。
EOS
```

第 1 章

概要

PyGeoNLP は自然言語テキストから地名を抽出し、地理的な属性を割り当てる GeoParser の Python ライブラリです。コマンドラインから **ジオパーズ処理** を呼び出してテキストを解析したり、地名抽出機能が必要なアプリケーションやサービスに組み込んで Python API を利用することができます。

リスト 1　Python API 使用例

```
>>> import pygeonlp.api
>>> print(pygeonlp.api.geoparse("NII は神保町駅から徒歩 7 分です。"))
[{'type': 'Feature', 'geometry': None, 'properties': {'surface': 'NII', 'node_type':
↪'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_form
↪': '*', 'pos': ' 名詞', 'prononciation': '', 'subclass1': ' 固有名詞', 'subclass2': ' 組織
↪', 'subclass3': '*', 'surface': 'NII', 'yomi': ''}}}, {'type': 'Feature', 'geometry':␣
↪None, 'properties': {'surface': ' は', 'node_type': 'NORMAL', 'morphemes': {'conjugated_
↪form': '*', 'conjugation_type': '*', 'original_form': ' は', 'pos': ' 助詞',
↪'prononciation': ' ワ', 'subclass1': ' 係助詞', 'subclass2': '*', 'subclass3': '*',
↪'surface': ' は', 'yomi': ' ハ'}}}, {'type': 'Feature', 'geometry': {'type': 'Point',
↪'coordinates': [139.757845, 35.6960275]}, 'properties': {'surface': ' 神保町駅', 'node_
↪type': 'GEOWORD', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*',
↪'original_form': ' 神保町駅', 'pos': ' 名 詞', 'prononciation': '', 'subclass1': ' 固有名詞
', 'subclass2': ' 地名語', 'subclass3': 'uN6ecI:神保町駅', 'surface': ' 神保町駅', 'yomi': '
↪'}, 'geoword_properties': {'body': ' 神保町', 'dictionary_id': 3, 'entry_id': '5WS6qh',
 'geolod_id': 'uN6ecI', 'hypernym': [' 東京都', '10 号線新宿線'], 'institution_type': ' 公営鉄
道', 'latitude': '35.6960275', 'longitude': '139.757845', 'ne_class': ' 鉄道施設/鉄道 駅',
↪'railway_class': ' 普通鉄道', 'suffix': [' 駅', ''], 'dictionary_identifier': 'geonlp:ksj-
↪station-N02'}}}, {'type': 'Feature', 'geometry': None, 'properties': {'surface': ' から
', 'node_type': 'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*',
↪'original_form': ' から', 'pos': ' 助詞', 'prononciation': ' カラ', 'subclass1': ' 格助詞',
↪'subclass2': ' 一般', 'subclass3': '*', 'surface': ' から', 'yomi': ' カラ'}}}, {'type':
↪'Feature', 'geometry': None, 'properties': {'surface': ' 徒歩', 'node_type': 'NORMAL',
```

(次のページに続く)

(前のページからの続き)

```
 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '徒歩',
↪'pos': '名詞', 'prononciation': 'トホ', 'subclass1': '一般', 'subclass2': '*',
 'subclass3': '*', 'surface': '徒歩', 'yomi': 'トホ'}}}, {'type': 'Feature', 'geometry':␣
↪None, 'properties': {'surface': '7', 'node_type': 'NORMAL', 'morphemes': {'conjugated_
↪form': '*', 'conjugation_type': '*', 'original_form': '*', 'pos': '名詞',
↪'prononciation': '', 'subclass1': '数', 'subclass2': '*', 'subclass3': '*', 'surface':
↪'7', 'yomi': ''}}}, {'type': 'Feature', 'geometry': None, 'properties': {'surface': '分
', 'node_type': 'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*',
↪'original_form': '分', 'pos': '名詞', 'prononciation': 'フン', 'subclass1': '接尾',
↪'subclass2': '助数詞', 'subclass3': '*', 'surface': '分', 'yomi': 'フ ン'}}}, {'type':
↪'Feature', 'geometry': None, 'properties': {'surface': 'です', 'node_type': 'NORMAL',
↪'morphemes': {'conjugated_form': '特殊・デス', 'conjugation_type': '基本形', 'original_
↪form': 'です', 'pos': '助動詞', 'prononciation': 'デス', 'subclass1': '*', 'subclass2':
↪'*', 'subclass3': '*', 'surface': 'です', 'yomi': 'デス'}}}, {'type': 'Feature',
↪'geometry': None, 'properties': {'surface': '。', 'node_type': 'NORMAL', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '。', 'pos': '記号',
↪'prononciation': '。', 'subclass1': '句点', 'subclass2': '*', 'subclass3': '*', 'surface
↪': '。', 'yomi': ' 。'}}}]
```

1.1 特徴

PyGeoNLP には以下のような特徴があります。

- ウェブサービスではないので、オフラインでも利用可能です
- 辞書を作って登録すれば独自の地名も抽出できます
- 住所ジオコーダーと連携すれば住所も抽出できます

1.2 処理の内容

PyGeoNLP は、まず自然文を MeCab で解析し、単語のリストを作ります。

次に、連続する単語を組み合わせてみて、**地名解析辞書**に登録されていれば**地名語**として抽出します。たとえば「神保町」＋「駅」は地名解析辞書の「神保町駅」と一致するので、地名語として一語にまとめます。

もし抽出した地名語に綴りが同じものが複数存在する場合には、フィルタを利用して絞り込んだり、他の地名語との関係を見てスコアを付けランキングするなどの地名解決処理を行ないます。

第 2 章

インストール手順

2.1 動作確認済み環境

pygeonlp は以下の環境で動作確認済みです。

- x86_64 アーキテクチャ / Ubuntu 22.04.3 LTS
- aarm64 アーキテクチャ / MacOS X Sonoma 14.1.1
- Windows Subsystem for Linux / Ubuntu 20.04.6 LTS
- x86_64, aarm64 アーキテクチャ / Docker desktop, Docker engine

環境ごとのインストール手順は、リンク先を参照してください。

- install_pygeonlp_ubuntu
- install_pygeonlp_macosx
- install_pygeonlp_docker
- install_pygeonlp_other

2.2 動作確認

インストールが完了したら動作確認を行います。**pygeonlp geoparse** コマンドを実行してください。

```
$ pygeonlp geoparse
/home/foo/geonlp/db に基本辞書セットをインストールしますか? (y/n):
```

初回実行時は上のように基本辞書セットをインストールするかどうか確認するメッセージが表示されますので、"y" を入力してください。

標準でインストールされる基本辞書は以下の 3 種類です。

- 日本の都道府県 (**geonlp:geoshape-pref**)
- 歴史的行政区域データセットβ版地名辞書 (**geonlp:geoshape-city**)
- 日本の鉄道駅（2019 年） (**geonlp:ksj-station-N02-2019**)

pygeonlp はデフォルトで環境変数 **HOME** の下の **geonlp/db** に辞書をインストールしたデータベースを作成します。インストール先を変更したい場合は setup のパラメータにインストール先ディレクトリを指定してください。

```
$ pygeonlp setup --db-dir=/home/foo/share/geonlp/db
```

または、環境変数 **GEONLP_DB_DIR** をセットしておくと、そのディレクトリを参照します。

$ export GEONLP_DB_DIR=/home/foo/share/geonlp/db $ pygeonlp setup

環境によって、付属の地名解析辞書が見つからず、「地名解析辞書がインストールされたディレクトリが見つかりません。」というエラーが発生する場合があります。

その場合は以下の手順でディレクトリを見つけ、**setup** のパラメータで指定してください。

- pip uninstall を実行してパッケージに含まれるファイルリストを確認します。**Proceed (y/n)?** には n と答えてください。

```
% pip uninstall pygeonlp
Uninstalling pygeonlp-1.0.0:
Would remove:
...
/opt/homebrew/pygeonlp_basedata/geoshape-pref.csv
...
Proceed (y/n)? n
```

- **geoshape-pref.csv** などが含まれているディレクトリをメモします。上の例では **/opt/homebrew/pygeonlp_basedata/** です。
- **setup** のパラメータとしてこのディレクトリを指定します。

```
$ pygeonlp setup /opt/homebrew/pygeonlp_basedata
```

辞書のインストールが終わると、テキストの入力待ちになります。地名を含む日本語のテキストを入力すると解析

結果が表示されます。終了したい場合は Ctrl+D で EOF コードを送信してください。

```
$ pygeonlp geoparse
/home/foo/geonlp/db に基本辞書セットをインストールしますか? (y/n): 
完了しました。
目黒駅は品川区上大崎にあります。
目黒駅　名詞,固有名詞,地名語,Xy26iV:目黒駅,*,*,目黒駅,,鉄道施設/鉄道駅,Xy26iV,目黒駅,139.
↪71566,35.632485
は　助詞,係助詞,*,*,*,*,は,ハ,ワ
品川区　名詞,固有名詞,地名語,kEAYBl:品川区,*,*,品川区,,市区町村,kEAYBl,品川区,139.73025000,
↪35.60906600
上大崎　名詞,固有名詞,地域,一般,*,*,上大崎,カミオオサキ,カミオーサキ
に　助詞,格助詞,一般,*,*,*,に,ニ,ニ
あり　動詞,自立,*,*,連用形,五段・ラ行,ある,アリ,アリ
ます　助動詞,*,*,*,基本形,特殊・マス,ます,マス,マス
。　記号,句点,*,*,*,*,。,。,。
EOS
品川駅は港区高輪にあります。
品川駅　名詞,固有名詞,地名語,jUUOco:品川駅,*,*,品川駅,,鉄道施設/鉄道駅,jUUOco,品川駅,139.
↪738535,35.628135
は　助詞,係助詞,*,*,*,*,は,ハ,ワ
港区　名詞,固有名詞,地名語,2CWYZ5:港区,*,*,港区,,市区町村,2CWYZ5,港区,139.75159900,
↪35.65807100
高輪　名詞,固有名詞,地域,一般,*,*,高輪,タカナワ,タカナワ
に　助詞,格助詞,一般,*,*,*,に,ニ,ニ
あり　動詞,自立,*,*,連用形,五段・ラ行,ある,アリ,アリ
ます　助動詞,*,*,*,基本形,特殊・マス,ます,マス,マス
。　記号,句点,*,*,*,*,。,。,。
EOS
```

2.3 pygeonlp のアンインストール

pygeonlp が不要になった場合は以下のコマンドでアンインストールできます。

```
$ pip uninstall pygeonlp
```

GDAL も不要な場合にはアンインストールしてください。

```
$ pip uninstall gdal
```

2.4 データベースの完全削除

地名語解析辞書をインストールすると、データベースディレクトリにいくつかのファイルを作成します。データベースディレクトリがどこに作成されるかは**データベースディレクトリ** を参照してください。

それ以外の場所は変更しませんので、全てのデータベースを削除したい場合はデータベースディレクトリごと消去してください。

(デフォルトの場合)

```
$ rm -r ~/geonlp
```

(環境変数 GEONLP_DB_DIR をセットした場合)

```
$ rm -r `echo $GEONLP_DB_DIR`
```

第 3 章

コマンドラインインタフェース

Pygeonlp の機能をコマンドラインから利用する方法を説明します。pygeonlp のコマンドは全て **pygeonlp** の後ろに実行したい機能を指定し、さらにパラメータが続きます。 **pygeonlp --help** でオンラインヘルプを表示します。

コマンドラインではジオパーズ処理の他、地名解析辞書の管理を行うことができます。

3.1 ジオパーズ処理

pygeonlp geoparse を実行すると、対話的に一行ずつテキストを入力し、ジオパーズした結果を出力します。

```
$ pygeonlp geoparse
NII は神保町にあります。
NII　　名詞，固有名詞，組織，*，*，*，*，，
は　　助詞，係助詞，*，*，*，*，は，ハ，ワ
神保町　名詞，固有名詞，地名語，uN6ecI：神保町駅，*，*，神保町，，　　鉄道施設/鉄道駅，uN6ecI，神保町駅，139.757845，35.6960275
に　　助詞，格助詞，一般，*，*，*，に，ニ，ニ
あり　　動詞，自立，*，*，連用形，五段・ラ行，ある，アリ，アリ
ます　　助動詞，*，*，*，基本形，特殊・マス，ます，マス，マス
。　　記号，句点，*，*，*，*，。，。，。
EOS
```

終了するには **Ctrl+D** を押して EOF を送信してください。

ファイルからパイプで接続して実行することもできます。

```
$ cat sample.txt | pygeonlp geoparse > result.txt
```

出力フォーマットは MeCab に似ています。

```
行 ::= 表層形 \t 形態素情報 \t 地理的情報
形態素情報 ::= 品詞,品詞細分類1,品詞細分類2,品詞細分類3,活用型,活用形,原形,読み,発音
地理的情報 ::= 固有名クラス,geolod_id,地名語,経度,緯度
```

表層形と形態素情報は MeCab の標準出力フォーマットと同じです。地名語または住所を抽出した場合、GeoNLP 固有の属性である地理的情報が付与されます。

--json オプションを指定すると、 Python API と同じ、より詳細な情報を含む GeoJSON のリスト形式で出力します。

```
$ echo "NII は神保町にあります。" | pygeonlp geoparse --json
[{"type": "Feature", "geometry": null, "properties": {"surface": "NII", "node_type":
↪"NORMAL", "morphemes": {"conjugated_form": "*", "conjugation_type": "*", "original_form
↪": "*", "pos": "名詞", "prononciation": "", "subclass1": "固有名詞", "subclass2": "組織",
 "subclass3": "*", "surface": "NII", "yomi": ""}}}, {"type": "Feature", "geometry": null,
 "properties": {"surface": "は", "node_type": "NORMAL", "morphemes": {"conjugated_form":
↪"*", "conjugation_type": "*", "original_form": "は", "pos": "助詞", "prononciation": "ワ
", "subclass1": "係助詞", "subclass2": "*", "subclass3": "*", "surface": "は", "yomi": "ハ
↪"}}}, {"type": "Feature", "geometry": {"type": "Point", "coordinates": [139.757845, 35.
↪6960275]}, "properties": {"surface": "神保町", "node_type": "GEOWORD", "morphemes": {
↪"conjugated_form": "*", "conjugation_type": "*", "original_form": "神保町", "pos": "名詞
↪", "prononciation": "", "subclass1": "固有名詞", "subclass2": "地名語", "subclass3":
↪"uN6ecI:神保町駅", "surface": "神保町", "yomi": ""}, "geoword_properties": {"body": "神保
町", "dictionary_id": 3, "entry_id": "5WS6qh", "geolod_id": "uN6ecI", "hypernym": ["東京都
", "10 号線新宿線"], "institution_type": "公営鉄道", "latitude": "35.6960275", "longitude":
↪"139.757845", "ne_class": "鉄道施設/鉄道駅", "railway_class": "普通鉄道", "suffix": ["駅",
 ""], "dictionary_identifier": "geonlp:ksj-station-N02"}}}, {"type": "Feature", "geometry
↪": null, "properties": {"surface": "に", "node_type": "NORMAL", "morphemes": {
↪"conjugated_form": "*", "conjugation_type": "*", "original_form": "に", "pos": "助詞",
↪"prononciation": "ニ", "subclass1": "格助詞", "subclass2": "一般", "subclass3": "*",
↪"surface": "に", "yomi": "ニ"}}}, {"type": "Feature", "geometry": null, "properties": {
↪"surface": "あり", "node_type": "NORMAL", "morphemes": {"conjugated_form": "五段・ラ行",
 "conjugation_type": "連用形", "original_form": "ある", "pos": "動詞", "prononciation": "ア
リ", "subclass1": "自立", "subclass2": "*", "subclass3": "*", "surface": "あり", "yomi":
↪"アリ"}}}, {"type": "Feature", "geometry": null, "properties": {"surface": "ます",
 "node_type": "NORMAL", "morphemes": {"conjugated_form": "特殊・マス", "conjugation_type":
↪"基本形", "original_form": "ます", "pos": "助動詞", "prononciation": "マス", "subclass1":
↪"*", "subclass2": "*", "subclass3": "*", "surface": "ます", "yomi": "マス"}}}, {"type":
↪"Feature", "geometry": null, "properties": {"surface": "。", "node_type": "NORMAL",
↪"morphemes": {"conjugated_form": "*", "conjugation_type": "*", "original_form": "。",
```

(次のページに続く)

(前のページからの続き)

```
 "pos": "記号", "prononciation": "。", "subclass1": "句点", "subclass2": "*", "subclass3":
↪"*", "surface": "。", "yomi": "。"}}}]
```

3.2 地名語検索

pygeonlp search で地名語をデータベースから検索します。

```
$ pygeonlp search 神保町
{'AGGwyc': {'body': '神保町', 'dictionary_id': 3, 'entry_id': 'Iy2jnd', 'geolod_id':
↪'AGGwyc', 'hypernym': ['東京地下鉄', '11 号線半蔵門線'], 'institution_type': '民営鉄道',
↪'latitude': '35.695975000000004', 'longitude': '139.757665', 'ne_class': '鉄道施設/鉄道駅
↪', 'railway_class': '普通鉄道', 'suffix': ['駅', ''], 'dictionary_identifier':
↪'geonlp:ksj-station-N02'}, 'c804ks': {'body': '神保町', 'dictionary_id': 3, 'entry_id':
↪'7t7dVr', 'geolod_id': 'c804ks', 'hypernym': ['東京都', '6 号線三田線'], 'institution_
↪type': '公営鉄道', 'latitude': '35.69489333333333', 'longitude': '139.75842833333334',
↪'ne_class': '鉄道施設/鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅', ''],
↪'dictionary_identifier': 'geonlp:ksj-station-N02'}, 'uN6ecI': {'body': '神保町',
↪'dictionary_id': 3, 'entry_id': '5WS6qh', 'geolod_id': 'uN6ecI', 'hypernym': ['東京都',
↪'10 号線新宿線'], 'institution_type': '公営鉄道', 'latitude': '35.6960275', 'longitude':
↪'139.757845', 'ne_class': '鉄道施設/鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅
↪', ''], 'dictionary_identifier': 'geonlp:ksj-station-N02'}}
```

3.3 基本辞書セットをインストール

pygeonlp setup で pygeonlp に同梱されている基本辞書セットをインストールします。

```
$ pygeonlp setup
```

このコマンドは基本辞書がアンインストールされていればインストールしなおします。

3.4 全ての辞書をアンインストール

pygeonlp clear-dictionaries を実行すると、インストールされている全ての辞書をアンインストールします。

```
$ pygeonlp clear-dictionaries
```

3.5 インストール済み辞書を一覧表示

pygeonlp list-dictionaries でインストール済みの地名解析辞書の一覧を表示します。

```
$ pygeonlp list-dictionaries
geonlp:geoshape-city : 歴史的行政区域データセットβ版地名辞書
- https://geonlp.ex.nii.ac.jp/dictionary/geoshape-city/
歴史的行政区域データセットβ版で構築した地名辞書です。1920年から2020年までの国土 数値情報「行政区域データ」に出現する市区町村をリスト化し、独自の固有IDを付与して公開しています。データセット構築の詳しい手法については、「歴史的行政区域データセットβ版」のウェブサイトをご覧ください。
...
```

先頭の **geonlp:geoshape-city** がこの辞書の識別子です。その後ろの「歴史的行政区域データセットβ版地名辞書」が辞書の名称です。2 行目の URL で辞書が公開されています。3 行目が辞書の概要説明です。

3.6 辞書の詳細情報を表示

一覧には概要しか表示されませんので、ライセンスや更新日時などを確認したい場合には **pygeonlp show-dictionary** で個別に辞書の詳細情報を表示してください。

```
$ pygeonlp show-dictionary geonlp:geoshape-city
{
"@context": "https://schema.org/",
"@type": "Dataset",
"alternateName": "",
"creator": [
    {
    "@type": "Organization",
    "name": "ROIS-DS 人文学オープンデータ共同利用センター",
    "sameAs": "http://codh.rois.ac.jp/"
    }
],
"dateModified": "2021-01-04T22:03:51+09:00",
```

(次のページに続く)

(前のページからの続き)


```
"description": "歴史的行政区域データセットβ版で構築した地名辞書です。1920 年か ら 2020 年までの国土数値情報「行政区域データ」に出現する市区町村をリスト化し、独自の固有 ID を付与して公開しています。データセット構築の詳しい手法については、「歴史的行政区域データセットβ版」のウェブサイトをご覧ください。",
"distribution": [
    {
    "@type": "DataDownload",
    "contentUrl": "http://agora.ex.nii.ac.jp/GeoNLP/dict/geoshape-city.csv",
    "encodingFormat": "text/csv"
    }
],
"identifier": [
    "geonlp:geoshape-city"
],
"isBasedOn": {
    "@type": "CreativeWork",
    "name": "歴史的行政区域データセットβ版",
    "url": "https://geoshape.ex.nii.ac.jp/city/"
},
"keywords": [
    "GeoNLP",
    "地名辞書"
],
"license": "https://creativecommons.org/licenses/by/4.0/",
"name": "歴史的行政区域データセットβ版地名辞書",
"size": "16421",
"spatialCoverage": {
    "@type": "Place",
    "geo": {
    "@type": "GeoShape",
    "box": "24.06092 123.004496 45.5566280626738 148.772556996888"
    }
},
"temporalCoverage": "../..",
"url": "https://geonlp.ex.nii.ac.jp/dictionary/geoshape-city/"
}
```

3.7 拡張辞書をインストール

基本辞書セット以外の地名解析辞書をインストールしたい場合、まず辞書が公開されているページの URL が必要です。

Google Dataset Search でキーワードに **geonlp** を指定すると、利用可能な辞書を簡単に見つけることができます。たとえば「geonlp 郵便局」で検索すると国土数値情報：郵便局データが見つかると思います。

この辞書をインストールするには **pygeonlp add-dictionary** に URL を指定します。

```
$ pygeonlp add-dictionary https://geonlp.ex.nii.ac.jp/dictionary/ksj-post-office/
```

3.8 辞書をアンインストール

インストール済みの地名解析辞書は、 **pygeonlp remove-dictionary** に辞書の識別子を指定すると個別にアンインストールできます。

```
$ pygeonlp remove-dictionary geonlp:post-office
```

第 4 章

Quickstart

ここでは pygeonlp を Python API を利用して操作する簡単な例を紹介します。まだ pygeonlp をインストールしていない場合は、**インストール手順** に従ってインストールしてください。

4.1 最小限のサンプル

pygeonlp を使って自然文テキストから地名を抽出する最小コードは次のようになります。

```
import pygeonlp.api
print(pygeonlp.api.geoparse("NII は神保町駅から徒歩 7 分です。"))
```

このコードは次の処理を行ないます。

1. *pygeonlp.api* モジュールを読み込みます。
2. *geoparse()* メソッドを呼んで、テキストを解析します。

4.2 実行結果

上のコードを実行すると、次のような結果が表示されます。

```
[{'type': 'Feature', 'geometry': None, 'properties': {'surface': 'NII', 'node_type':
↪'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_form
↪': '*', 'pos': ' 名詞', 'prononciation': '', 'subclass1': ' 固有名詞', 'subclass2': ' 組織
↪', 'subclass3': '*', 'surface': 'NII', 'yomi': ''}}}, {'type': 'Feature', 'geometry':␣
↪None, 'properties': {'surface': ' は', 'node_type': 'NORMAL', 'morphemes': {'conjugated_
↪form': '*', 'conjugation_type': '*', 'original_form': ' は', 'pos': ' 助詞',
↪'prononciation': ' ワ', 'subclass1': ' 係助詞', 'subclass2': '*', 'subclass3': '*',
↪'surface': ' は', 'yomi': ' ハ'}}}, {'type': 'Feature', 'geometry': {'type': 'Point',
```

(次のページに続く)

(前のページからの続き)

```
↪'coordinates': [139.757845, 35.6960275]}, 'properties': {'surface': '神保町駅', 'node_
↪type': 'GEOWORD', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*',
 'original_form': '神保町駅', 'pos': '名詞', 'prononciation': '', 'subclass1': '固有名詞',
 'subclass2': '地名語', 'subclass3': '82wiE0:神保町駅', 'surface': '神保町駅', 'yomi': ''},
 'geoword_properties': {'body': '神保町', 'dictionary_id': 3, 'entry_id':
↪'2891e10e9314a0b378fac6aace6d2a7f', 'geolod_id': '82wiE0', 'hypernym': ['東京都', '10号
線新宿線'], 'institution_type': '公営鉄道', 'latitude': '35.6960275', 'longitude': '139.
↪757845', 'ne_class': '鉄道施設/鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅', '
↪'], 'dictionary_identifier': 'geonlp:ksj-station-N02-2019'}}}, {'type': 'Feature',
 'geometry': None, 'properties': {'surface': 'から', 'node_type': 'NORMAL', 'morphemes':
↪{'conjugated_form': '*', 'conjugation_type': '*', 'original_form': 'から', 'pos': '助詞
↪', 'prononciation': 'カラ', 'subclass1': '格助詞', 'subclass2': '一般', 'subclass3': '*
↪', 'surface': 'から', 'yomi': 'カラ'}}}, {'type': 'Feature', 'geometry': None,
↪'properties': {'surface': '徒歩', 'node_type': 'NORMAL', 'morphemes': {'conjugated_form
↪': '*', 'conjugation_type': '*', 'original_form': '徒歩', 'pos': '名詞', 'prononciation
↪': 'トホ', 'subclass1': '一般', 'subclass2': '*', 'subclass3': '*', 'surface': '徒歩',
 'yomi': 'トホ'}}}, {'type': 'Feature', 'geometry': None, 'properties': {'surface': '7',
↪'node_type': 'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*',
↪'original_form': '*', 'pos': '名詞', 'prononciation': '', 'subclass1': '数', 'subclass2
↪': '*', 'subclass3': '*', 'surface': '7', 'yomi': ''}}}, {'type': 'Feature', 'geometry
↪': None, 'properties': {'surface': '分', 'node_type': 'NORMAL', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '分', 'pos': '名詞',
↪'prononciation': 'フン', 'subclass1': '接尾', 'subclass2': '助数詞', 'subclass3': '*',
 'surface': '分', 'yomi': 'フン'}}}, {'type': 'Feature', 'geometry': None, 'properties':
↪{'surface': 'です', 'node_type': 'NORMAL', 'morphemes': {'conjugated_form': '特殊・デス
↪', 'conjugation_type': '基本形', 'original_form': 'です', 'pos': '助動詞',
↪'prononciation': 'デス', 'subclass1': '*', 'subclass2': '*', 'subclass3': '*', 'surface
↪': 'です', 'yomi': 'デス'}}}, {'type': 'Feature', 'geometry': None, 'properties': {
↪'surface': '。', 'node_type': 'NORMAL', 'morphemes': {'conjugated_form': '*',
↪'conjugation_type': '*', 'original_form': '。', 'pos': '記号', 'prononciation': '。',
 'subclass1': '句点', 'subclass2': '*', 'subclass3': '*', 'surface': '。', 'yomi': '。'}}}
↪]
```

このままでは見にくいので、 JSON で整形表示するように少し修正します。

```
import json
import pygeonlp.api
parsed = pygeonlp.api.geoparse("NII は神保町駅から徒歩 7 分です。")
print(json.dumps(parsed, indent=2, ensure_ascii=False))
```

実行結果は次のようになります。テキストが単語に分割され、それぞれの単語の品詞情報などが付与されているのが分かると思います。


```
[
  {
    "type": "Feature",
    "geometry": null,
    "properties": {
      "surface": "NII",
      "node_type": "NORMAL",
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "組織",
        "subclass3": "*",
        "surface": "NII",
        "yomi": ""
      }
    }
  },
  {
    "type": "Feature",
    "geometry": null,
    "properties": {
      "surface": "は",
      "node_type": "NORMAL",
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "は",
        "pos": "助詞",
        "prononciation": "ワ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "は",
```


(次のページに続く)

(前のページからの続き)


```
        "yomi": "ハ"
      }
    }
  },
  {
    "type": "Feature",
    "geometry": {
      "type": "Point",
      "coordinates": [
        139.757845,
        35.6960275
      ]
    },
    "properties": {
      "surface": "神保町駅",
      "node_type": "GEOWORD",
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "神保町駅",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "82wiE0:神保町駅",
        "surface": "神保町駅",
        "yomi": ""
      },
      "geoword_properties": {
        "body": "神保町",
        "dictionary_id": 3,
        "entry_id": "2891e10e9314a0b378fac6aace6d2a7f",
        "geolod_id": "82wiE0",
        "hypernym": [
          "東京都",
          "10 号線新宿線"
        ],
        "institution_type": "公営鉄道",
        "latitude": "35.6960275",
```


(次のページに続く)

(前のページからの続き)


```
        "longitude": "139.757845",
        "ne_class": "鉄道施設/鉄道駅",
        "railway_class": "普通鉄道",
        "suffix": [
          "駅",
          ""
        ],
        "dictionary_identifier": "geonlp:ksj-station-N02-2019"
      }
    }
  },
  {
    "type": "Feature",
    "geometry": null,
    "properties": {
      "surface": "から",
      "node_type": "NORMAL",
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "から",
        "pos": "助詞",
        "prononciation": "カラ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "から",
        "yomi": "カラ"
      }
    }
  },
  {
    "type": "Feature",
    "geometry": null,
    "properties": {
      "surface": "徒歩",
      "node_type": "NORMAL",
      "morphemes": {
        "conjugated_form": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "conjugation_type": "*",
        "original_form": "徒歩",
        "pos": "名詞",
        "prononciation": "トホ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "徒歩",
        "yomi": "トホ"
      }
    }
  },
  {
    "type": "Feature",
    "geometry": null,
    "properties": {
      "surface": "7",
      "node_type": "NORMAL",
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "7",
        "yomi": ""
      }
    }
  },
  {
    "type": "Feature",
    "geometry": null,
    "properties": {
      "surface": "分",
      "node_type": "NORMAL",
      "morphemes": {
```


(次のページに続く)

(前のページからの続き)


```
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "分",
        "pos": "名詞",
        "prononciation": "フン",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "分",
        "yomi": "フン"
      }
    }
  },
  {
    "type": "Feature",
    "geometry": null,
    "properties": {
      "surface": "です",
      "node_type": "NORMAL",
      "morphemes": {
        "conjugated_form": "特殊・デス",
        "conjugation_type": "基本形",
        "original_form": "です",
        "pos": "助動詞",
        "prononciation": "デス",
        "subclass1": "*",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "です",
        "yomi": "デス"
      }
    }
  },
  {
    "type": "Feature",
    "geometry": null,
    "properties": {
      "surface": "。",
      "node_type": "NORMAL",
```


(次のページに続く)

(前のページからの続き)

```
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "。",
        "pos": "記号",
        "prononciation": "。",
        "subclass1": "句点",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "。",
        "yomi": "。"
      }
    }
  }
]
```

4.3 地名語ノード

テキストを単語に分割するのは形態素解析器、または POS Tagger と呼ばれるツールに共通の機能です。 pygeonlp は、分割した単語（またはその組み合わせ）から地名解析辞書に登録されている地名語を見つけ、経緯度などを付け加える機能を持っている点が特徴です。

解析結果のうち、 node_type が GEOWORD となっている部分が地名語です。

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [
      139.757845,
      35.6960275
    ]
  },
  "properties": {
    "surface": "神保町駅",
    "node_type": "GEOWORD",
    "morphemes": {
      "conjugated_form": "*",
```

(次のページに続く)

(前のページからの続き)


```
      "conjugation_type": "*",
      "original_form": "神保町駅",
      "pos": "名詞",
      "prononciation": "",
      "subclass1": "固有名詞",
      "subclass2": "地名語",
      "subclass3": "82wiE0:神保町駅",
      "surface": "神保町駅",
      "yomi": ""
    },
    "geoword_properties": {
      "body": "神保町",
      "dictionary_id": 3,
      "entry_id": "2891e10e9314a0b378fac6aace6d2a7f",
      "geolod_id": "82wiE0",
      "hypernym": [
        "東京都",
        "10 号線新宿線"
      ],
      "institution_type": "公営鉄道",
      "latitude": "35.6960275",
      "longitude": "139.757845",
      "ne_class": "鉄道施設/鉄道駅",
      "railway_class": "普通鉄道",
      "suffix": [
        "駅",
        ""
      ],
      "dictionary_identifier": "geonlp:ksj-station-N02-2019"
    }
  }
}
```


この地名語部分は GeoJSON の Feature 形式になっていますので、この出力結果をテキストファイルに保存して GIS アプリケーションで開けば、地図上にプロットすることができます。

簡単にテストしたければ geojson.io にコピーした GeoJSON を貼り付ければ、神保町駅にマーカーがプロットされることを確認できます。

4.4 より高度な使い方

基本的な pygeonlp の使い方は以上です。

より進んだ使い方を知りたい方は、関連する説明へお進みください。

- NEologd と連携して固有表現の抽出精度を上げたい
  - *NEologd* **連携** へ
- 住所文字列を住所として解析したい
  - **住所ジオコーダー連携** へ
- 別の場所にある同じ名前の地名が抽出されてしまうのでチューニングしたい
  - **解析方法のチューニング** へ
- より高度な解析を行うモジュールを開発したい
  - **カスタム解析モジュールの開発** へ
- 抽出したい地名が辞書に載っていないので、独自の地名解析辞書を作りたい
  - 辞書制作者向け資料 へ

第 5 章

NEologd 連携

pygeonlp を NEologd (Neologism dictionary for MeCab) と連携することで、テキスト中の固有表現の抽出精度を改善できます。

NEologd のインストール手順は公式サイト を参照してください。

以下に手順の一例を示します。

```
$ mkdir -p ~/github
$ cd ~/github
$ git clone --depth 1 https://github.com/neologd/mecab-ipadic-neologd.git
$ cd mecab-ipadic-neologd
$ ./bin/install-mecab-ipadic-neologd --prefix ~/neologd -n -a
```

上の例では **~/github/mecab-ipadic-neologd/** に GitHub リポジトリを clone し、**~/neologd/** に全部入り (-a) の最新辞書 (-n) をインストールします。

5.1 pygeonlp から NEologd を利用する

pygeonlp で mecab-ipadic-NEologd を MeCab システム辞書として利用するには、環境変数 **GEONLP_MECAB_DIC_DIR** に neologd をインストールしたディレクトリを指定してください。

```
$ export GEONLP_MECAB_DIC_DIR=~/neologd
$ echo "国立情報学研究所は千代田区にあります。" | pygeonlp geoparse
国立情報学研究所　　名詞,固有名詞,組織,*,*,*,国立情報学研究所,コクリツジョウホウガクケンキュウジョ,コクリツジョーホーガクケンキュージョ
は　　助詞,係助詞,*,*,*,*,は,ハ,ワ
千代田区　　名詞,固有名詞,地名語,WWIY7G:千代田区,*,,千代田区,,　　市区町村,WWIY7G,千代田区,139.75363400,35.69400300
```

(次のページに続く)

(前のページからの続き)

```
に　　助詞,格助詞,一般,*,*,*,に,ニ,ニ
あり　　動詞,自立,*,*,連用形,五段・ラ行,ある,アリ,アリ
ます　　助動詞,*,*,*,基本形,特殊・マス,ます,マス,マス
。　　記号,句点,*,*,*,*,。,。,。
EOS
```

「国立情報学研究所」が一語の固有名詞として抽出されるようになります。

GEONLP_MECAB_DIC_DIR を指定しない場合はデフォルトのシステム辞書を利用します。

```
$ unset GEONLP_MECAB_DIC_DIR
$ echo "国立情報学研究所は千代田区にあります。" | pygeonlp geoparse
国立　　名詞,一般,*,*,*,*,国立,コクリツ,コクリツ
情報　　名詞,一般,*,*,*,*,情報,ジョウホウ,ジョーホー
学　　名詞,接尾,一般,*,*,*,学,ガク,ガク
研究所　名詞,一般,*,*,*,*,研究所,ケンキュウジョ,ケンキュージョ
は　　助詞,係助詞,*,*,*,*,は,ハ,ワ
千代田区　　名詞,固有名詞,地名語,WWIY7G:千代田区,*,*,千代田区,,　　市区町村,WWIY7G,千代田
区,139.75363400,35.69400300
に　　助詞,格助詞,一般,*,*,*,に,ニ,ニ
あり　　動詞,自立,*,*,連用形,五段・ラ行,ある,アリ,アリ
ます　　助動詞,*,*,*,基本形,特殊・マス,ます,マス,マス
。　　記号,句点,*,*,*,*,。,。,。
EOS
```

環境変数を使わずに Python API で実行時に MeCab システム辞書を指定するには、*pygeonlp.api.init()* の解析オプション **system_dic_dir** で NEologd のカスタムシステム辞書のパスを指定してください。

```
$ python
>>> import pygeonlp.api as api
>>> import os
>>> api.init(system_dic_dir=os.path.join(os.environ['HOME'], 'neologd'))
>>> [(x['properties']['surface'], x['properties']['node_type']) for x in api.geoparse('国
立情報学研究所は千代田区にあります。')]
[('国立情報学研究所', 'NORMAL'), ('は', 'NORMAL'), ('千代田区', 'GEOWORD'), ('に', 'NORMAL
↪'), ('あり', 'NORMAL'), ('ます', 'NORMAL'), ('。', 'NORMAL')]
```

第 6 章

住所ジオコーダー連携

6.1 住所辞書データのインストール

pygeonlp を住所ジオコーダー jageocoder と連携することで、テキスト中の住所を地名語ではなく住所として認識できます。

jageocoder の住所辞書はサイズが大きいため、 pygeonlp をインストールしても自動的にはインストールされません。住所を抽出したい場合は以下の手順で住所辞書をインストールしてください。

- 利用する住所辞書データをダウンロードする

  Jageocoder データファイル一覧に最新の住所辞書データがあります。利用したい辞書データ（zip ファイル）をダウンロードしてください。

- 住所辞書をインストールする

  jageocoder install-dictionary コマンドでダウンロードした辞書データをインストールします。

```
$ jageocoder install-dictionary <ダウンロードした zip ファイルのパス>
```

- 動作確認する

  pygeonlp geoparse で住所を含むテキストを解析し、住所部分が抽出されることを確認します。

```
$ echo "東京都庁は新宿区西新宿２－８にあります。" | pygeonlp geoparse
東京　　名詞，固有名詞，地名語,0q60k0:東京駅,*,,東京，トウキョウ，トウキョウ　　鉄道施設/
↪鉄道駅,0q60k0，東京駅,139.766685,35.680965
都庁　　名詞，一般,*,*,*,*,都庁，トチョウ，トチョー
は　　助詞，係助詞,*,*,*,*,は，ハ，ワ
新宿区西新宿２－８　　名詞，固有名詞，住所，街区・地番,*,*,東京都新宿区西新宿二丁目 8 番，ニハ
チ，ニハチ　　住所,,東京都新宿区西新宿二丁目 8 番,139.6917724609375,35.68962860107422
に　　助詞，格助詞，一般,*,*,*,に，ニ，ニ
```

(次のページに続く)

(前のページからの続き)

```
あり　　動詞,自立,*,*,連用形,五段・ラ行,ある,アリ,アリ
ます　　助動詞,*,*,*,基本形,特殊・マス,ます,マス,マス
。　　記号,句点,*,*,*,*,。,。,。
EOS
```

6.2 一時的に住所を解析しない

辞書データがインストールされていれば、住所ジオコーダーは自動的に利用されます。何らかの理由で住所を解析したくない時は、環境変数 **JAGEOCODER_DB2_DIR** に **false** など **住所辞書をインストールしたディレクトリ名以外**をセットしてください。住所辞書が見つからなくなるので、住所を解析しません。

```
$ export JAGEOCODER_DB2_DIR=false
$ echo "東京都庁は新宿区西新宿２－８にあります。" | pygeonlp geoparse
東京　　名詞,固有名詞,地名語,yaOcgu:東京駅,*,,東京,トウキョウ,トウキョウ　　鉄道施設/鉄道駅
,yaOcgu,東京駅,139.76479999999998,35.681934999999996
都庁　　名詞,一般,*,*,*,*,都庁,トチョウ,トチョー
は　　助詞,係助詞,*,*,*,*,は,ハ,ワ
新宿区　名詞,固有名詞,地名語,5q2IUM:新宿区,*,*,新宿区,,市区町村,5q2IUM,新宿区,139.70346300,
↪35.69389000
西新宿　名詞,固有名詞,地名語,ZgyYyS:西新宿駅,*,,西新宿,,　　鉄道施設/鉄道駅,ZgyYyS,西新宿
駅,139.69256000000001,35.694514999999996
２　　名詞,数,*,*,*,*,２,ニ,ニ
－　　記号,一般,*,*,*,*,*,,
８　　名詞,数,*,*,*,*,８,ハチ,ハチ
に　　助詞,格助詞,一般,*,*,*,に,ニ,ニ
あり　　動詞,自立,*,*,連用形,五段・ラ行,ある,アリ,アリ
ます　　助動詞,*,*,*,基本形,特殊・マス,ます,マス,マス
。　　記号,句点,*,*,*,*,。,。,。
EOS
```

環境変数を使わずに Python API で実行時に住所辞書を使わないよう指定するには、*pygeonlp.api.init()* を呼びだす時に **jageocoder** オプションに False をセットしてください。

```
>>> import pygeonlp.api as api
>>> api.init(jageocoder=False)
>>> api.geoparse('NII は千代田区一ツ橋 2-1-2 にあります。')
[{'type': 'Feature', 'geometry': None, 'properties': {'surface': 'NII', 'node_type':
↪'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_form
```

(次のページに続く)

(前のページからの続き)


```
↪': '*', 'pos': ' 名詞', 'prononciation': '', 'subclass1': ' 固有名詞', 'subclass2': ' 組織
↪', 'subclass3': '*', 'surface': 'NII', 'yomi': ''}}}, {'type': 'Feature', 'geometry':␣
↪None, 'properties': {'surface': ' は', 'node_type': 'NORMAL', 'morphemes': {'conjugated_
↪form': '*', 'conjugation_type': '*', 'original_form': ' は', 'pos': ' 助詞',
↪'prononciation': ' ワ', 'subclass1': ' 係助詞', 'subclass2': '*', 'subclass3': '*',
↪'surface': ' は', 'yomi': ' ハ'}}}, {'type': 'Feature', 'geometry': {'type': 'Point',
 'coordinates': [139.758148, 35.692332]}, 'properties': {'surface': ' 千代田区一ツ橋 2-1-',
↪'node_type': 'ADDRESS', 'morphemes': [{'surface': ' 千代田区', 'node_type': 'GEOWORD',
 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_form': ' 千代田区
↪', 'pos': ' 名詞', 'prononciation': '', 'subclass1': ' 固有名詞', 'subclass2': ' 地名語',
 'subclass3': 'WWIY7G:千代田区', 'surface': ' 千代田区', 'yomi': ''}, 'geometry': {'type':
↪'Point', 'coordinates': [139.753634, 35.694003]}, 'prop': {'address': ' 東京都千代田区',
↪'body': ' 千代田', 'body_variants': ' 千代田', 'code': {}, 'countyname': '', 'countyname_
↪variants': '', 'dictionary_id': 1, 'entry_id': '13101A1968', 'geolod_id': 'WWIY7G',
 'hypernym': [' 東京都'], 'latitude': '35.69400300', 'longitude': '139.75363400', 'ne_class
↪': ' 市区町村', 'prefname': ' 東京都', 'prefname_variants': ' 東京都', 'source': '1/千代田区
役所/千代田区九段南1-2-1/P34-14_13.xml', 'suffix': [' 区'], 'valid_from': '', 'valid_to': '
↪', 'dictionary_identifier': 'geonlp:geoshape-city'}}, {'surface': ' 一ツ橋', 'node_type
↪': 'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_
↪form': ' 一ツ橋', 'pos': ' 名詞', 'prononciation': ' ヒトツバシ', 'subclass1': ' 固有名詞',
↪'subclass2': ' 地域', 'subclass3': ' 一般', 'surface': ' 一ツ橋', 'yomi': ' ヒトツバシ'},
↪'geometry': None, 'prop': None}, {'surface': '2', 'node_type': 'NORMAL', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '*', 'pos': ' 名詞',
↪'prononciation': '', 'subclass1': ' 数', 'subclass2': '*', 'subclass3': '*', 'surface':
↪'2', 'yomi': ''}, 'geometry': None, 'prop': None}, {'surface': '-', 'node_type':
↪'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_form
↪': '*', 'pos': ' 名詞', 'prononciation': '', 'subclass1': ' サ変接続', 'subclass2': '*',
↪'subclass3': '*', 'surface': '-', 'yomi': ''}, 'geometry': None, 'prop': None}, {
↪'surface': '1', 'node_type': 'NORMAL', 'morphemes': {'conjugated_form': '*',
↪'conjugation_type': '*', 'original_form': '*', 'pos': ' 名詞', 'prononciation': '',
↪'subclass1': ' 数', 'subclass2': '*', 'subclass3': '*', 'surface': '1', 'yomi': ''},
↪'geometry': None, 'prop': None}, {'surface': '-', 'node_type': 'NORMAL', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '*', 'pos': ' 名詞',
↪'prononciation': '', 'subclass1': ' サ変接続', 'subclass2': '*', 'subclass3': '*',
↪'surface': '-', 'yomi': ''}, 'geometry': None, 'prop': None}], 'address_properties': {
↪'id': 11460296, 'name': '1番', 'x': 139.758148, 'y': 35.692332, 'level': 7, 'note':␣
↪None, 'fullname': [' 東京都', ' 千代田区', ' 一ツ橋', ' 二丁目', '1番']}}}, {'type':
↪'Feature', 'geometry': None, 'properties': {'surface': '2', 'node_type': 'NORMAL',
↪'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '*',
```


(次のページに続く)

(前のページからの続き)

```
 'pos': '名詞', 'prononciation': '', 'subclass1': '数', 'subclass2': '*', 'subclass3': '*
↪', 'surface': '2', 'yomi': ''}}}, {'type': 'Feature', 'geometry': None, 'properties': {
↪'surface': 'に', 'node_type': 'NORMAL', 'morphemes': {'conjugated_form': '*',
↪'conjugation_type': '*', 'original_form': 'に', 'pos': '助詞', 'prononciation': 'ニ',
 'subclass1': '格助詞', 'subclass2': '一般', 'subclass3': '*', 'surface': 'に', 'yomi': '
ニ'}}}, {'type': 'Feature', 'geometry': None, 'properties': {'surface': 'あり', 'node_type
↪': 'NORMAL', 'morphemes': {'conjugated_form': '五段・ラ行', 'conjugation_type': '連用形
', 'original_form': 'ある', 'pos': '動詞', 'prononciation': 'アリ', 'subclass1': '自立',
↪'subclass2': '*', 'subclass3': '*', 'surface': 'あり', 'yomi': 'アリ'}}}, {'type':
↪'Feature', 'geometry': None, 'properties': {'surface': 'ます', 'node_type': 'NORMAL',
↪'morphemes': {'conjugated_form': '特殊・マス', 'conjugation_type': '基本形', 'original_
↪form': 'ます', 'pos': '助動詞', 'prononciation': 'マス', 'subclass1': '*', 'subclass2':
↪'*', 'subclass3': '*', 'surface': 'ます', 'yomi': 'マス'}}}, {'type': 'Feature',
↪'geometry': None, 'properties': {'surface': '。', 'node_type': 'NORMAL', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '。', 'pos': '記号',
↪'prononciation': '。', 'subclass1': '句点', 'subclass2': '*', 'subclass3': '*', 'surface
↪': '。', 'yomi': '。'}}}]
```

このサンプルコードは以下の処理を行ないます。

1. *pygeonlp.api* モジュールを import します。
2. *pygeonlp.api.init()* を呼んでデフォルトワークフローを初期化します。
3. *pygeonlp.api.geoparse()* を実行します。

住所ノードは **node_type** が ADDRESS になります。また、住所ノードは地名語ノードと同じように、 JSON エンコードすれば GeoJSON Feature オブジェクトになります。

住所辞書がインストールされていない時に **jageocoder=True** を指定すると、ParseError 例外が発生します。住所解析が必須の場合には True を、どちらでも構わない場合は **jageocoder** パラメータを省略してください。

6.3 処理中に切り替えたい場合

処理中にジオコーダーの利用をオン・オフしたい場合は、次のように *Parser* クラスの *set_jageocoder()* を直接呼び出して明示的に切り替えることもできます。

住所解析を行ないたい場合 ::

```
>>> api.default_workflow().parser.set_jageocoder(True)
```

住所解析を行ないたくない場合 ::

```
>>> api.default_workflow().parser.set_jageocoder(False)
```

第 7 章

ウェブサービス機能

JavaScript で構築されたウェブアプリケーションのフロントエンドや、Python 以外の言語で開発したアプリケーションから、地名語辞書を検索したりテキストのジオパージング処理を行ないたい場合があります。そんなときは pygeonlp の解析・検索機能をウェブサービスとして提供するバックエンドサーバを構築すれば、通信によってアプリケーション側から利用できます。

7.1 インストール手順

ウェブサービス機能は pygeonlp に含まれているため、**インストール手順** に従って pygeonlp をインストールしてください。

また、ウェブサービス機能でのみ利用する Python パッケージ Flask JSON-RPC は別途インストールする必要があります。

Flask JSON-RPC をインストールするには pip コマンドを利用します。

```
pip3 install flask-jsonrpc
```

7.1.1 インストール後の作業

データベースの作成

もしまだの場合、 pygeonlp コマンドでデータベースを作成してください。

```
$ pygeonlp setup
```

で基本辞書セットがインストールされたデータベースを作成します。

データベースの管理方法については **基本辞書セットをインストール** 以降の説明を参照してください。

7.1.2 アンインストール

ウェブサービス機能が不要になった場合は Flask JSON-RPC をアンインストールしてください。

```
$ pip3 uninstall flask-jsonrpc
```

Pygeonlp もアンインストールする場合は *pygeonlp* **のアンインストール**の手順に従ってください。

7.2 サーバ実行

7.2.1 テストサーバ起動

テストサーバを起動するには次のコマンドを実行します。

```
$ python3 -m flask --app=pygeonlp.webapi.app run --port=5000
```

他のマシンからアクセスする必要がある場合は `--host=0.0.0.0` も指定してください。

```
$ python3 -m flask --app=pygeonlp.webapi.app run --port=5000 --host=0.0.0.0
```

テストサーバは `Ctrl+C` で終了します。

7.2.2 動作テスト

Flask JSON-RPC の web browsable API 機能を利用して、ブラウザ上で WebAPI の確認とテストを行なうことができます。

サービス起動後、**Web brosable API の URL** (デフォルトでは http://127.0.0.1:5000/api/browse/) を開き、動作確認したいメソッドを左メニューから選択します。

パラメータを入力するダイアログが表示されますので、必要なパラメータを入力して **Save** ボタンを押すと、送信されたリクエストが Request 欄に、処理結果が Response 欄に表示されます。

7.2.3 プロダクションサーバ起動

Flask はデバッグ・テスト目的のサーバのため、実際に運用する際には Gunicorn など他の WSGI サーバを利用する必要があります。

たとえば Gunicorn を利用する場合は次のように実行してください。

```
$ pip3 install gunicorn
$ gunicorn pygeonlp.webapi.app:app --bind=127.0.0.1:5000
```

--bind はこのサーバにアクセスできるホストとポートを指定します。どこからでもアクセスできるサービスをポート 8000 で公開したい場合は**--bind=0.0.0.0:8000** のように指定してください。

7.2.4 サーバ設定

Pygeonlp の環境変数を利用して WebAPI サーバの設定を変更できます。詳細は **環境変数** を参照してください。

デフォルト以外のディレクトリにデータベースを作成した場合はサーバを起動する前に環境変数 **GEONLP_DB_DIR** にデータベースを作成したディレクトリのパスをセットしてください。

```
$ export GEONLP_DB_DIR=/usr/local/share/geonlp/db/
```

MeCab システム辞書として NEologd などデフォルトの IPADIC 以外の辞書を利用する場合、サーバを起動する前に環境変数 **GEONLP_MECAB_DIC_DIR** に辞書ディレクトリのパスをセットしてください。

```
$ export MECAB_DIC_DIR=$HOME/mecab-ipadic-neologd/
```

詳細は *NEologd* **連携** を参照してください。

住所辞書をインストールしたいディレクトリを環境変数 **JAGEOCODER_DB2_DIR** に設定して、Jageocoder の住所辞書をインストールしてからサーバを起動してください。

```
$ export JAGEOCODER_DB2_DIR=$HOME/jageocoder/db/
```

詳細は **住所ジオコーダー連携** を参照してください。

7.3 Docker で実行

ウェブサービスのサーバを Docker で起動する方法について説明します。

7.3.1 イメージの作成

空のフォルダを作成し、その中にテキストエディタで以下の内容を含む Dockerfile を作成します。

```
FROM osgeo/gdal:ubuntu-full-3.6.3
# このイメージは "Ubuntu 22.04.2 LTS" を拡張しています。

ENV JAGEOCODER_DB2_DIR /opt/db2

# pygeonlp に必要なライブラリ・パッケージをインストールします。
RUN apt-get update && apt-get install -y \
    libmecab-dev \
    mecab-ipadic-utf8 \
    libboost-all-dev \
    libsqlite3-dev \
    curl \
    python3 \
    python3-dev \
    python3-pip

# pygeonlp と基本辞書セットをインストールします。
RUN python3 -m pip install pygeonlp && pygeonlp setup

# ここから WebAPI サーバ固有の処理（Flask のインストールとサーバ起動）
RUN python3 -m pip install flask-jsonrpc
ENV GEONLP_MECAB_DIC_DIR=/var/lib/mecab/dic/ipadic-utf8
ENV FLASK_APP=pygeonlp.webapi.app

# 住所ジオコーダをインストールしたイメージを作りたい場合は以下のコメントを外してください。
# RUN curl https://www.info-proto.com/static/jageocoder/latest/gaiku_all_v21.zip \
#     -o /opt/gaiku_all_v21.zip && \
#     jageocoder install-dictionary /opt/gaiku_all_v21.zip && \
#     rm /opt/gaiku_all_v21.zip

# NEologd を辞書として利用したい場合は以下のコメントを外してください。
# ENV GEONLP_MECAB_DIC_DIR=/neologd
```

(次のページに続く)

(前のページからの続き)

```
# RUN apt-get install -y git mecab
# RUN cd /tmp \
#     && git clone --depth 1 https://github.com/neologd/mecab-ipadic-neologd.git \
#     && cd mecab-ipadic-neologd \
#     && ./bin/install-mecab-ipadic-neologd --prefix /neologd -n -a -u -y \
#     && cd / \
#     && rm -r /tmp/mecab-ipadic-neologd

# WebAPI サーバを起動します。
CMD ["/bin/bash", "-c", "python -m flask run -h 0.0.0.0 -p 5000"]
```

jageocoder や neologd を利用したい場合はコメントを外してください。

Docker ファイルを作成したディレクトリで、docker build を実行してイメージを作成します。

```
$ docker build -t pygeonlp_webapi_image .
```

これで *pygeonlp_webapi_image* というタグ名を持つイメージが作成されます。

7.3.2 サーバコンテナを起動

作成したイメージから、docker run でサーバを実行するコンテナを起動します。

```
$ docker run --name pygeonlp_webapi -d -p 5000:5000 pygeonlp_webapi_image
```

-d オプションはデタッチドモード (コンテナをバックグラウンドで実行) を指定し、**-p** オプションは Docker のホストのポートをコンテナ内のポートに接続します。**--name** オプションでコンテナ名を指定しているので、*pygeonlp_webapi* という名前を持つコンテナを生成します。

この状態で WebAPI サーバが起動しているので、http://localhost:5000/api/browse にアクセスして WebAPI ブラウザが動いていることを確認します。

7.3.3 サーバコンテナを終了

サーバコンテナはディタッチドモードで動き続けるため、終了するときには docker stop で停止します。パラメータはコンテナ名です。

```
$ docker stop pygeonlp_webapi
```

終了したサーバコンテナは docker start で再起動できます。

```
$ docker start pygeonlp_webapi
```

7.3.4 コンテナとイメージを削除

不要になったコンテナは docker rm で削除できます。パラメータはコンテナ名です。

```
$ docker rm pygeonlp_webapi
```

イメージも不要になった場合は、docker rmi で削除します。パラメータはイメージのタグ名です。

```
$ docker rmi pygeonlp_webapi_image
```

7.4 WebAPI の使い方

pygeonlp.webapi は JSON-RPC 2.0 を利用します。

サービスのエンドポイント /api に JSON リクエストメッセージを POST し、JSON レスポンスを受け取ります。

```
$ curl -X POST -H "Content-Type: application/json" \
  -d '{"jsonrpc":"2.0", "method": "geonlp.parse", \
  "params": {"sentence": "NII は神保町駅から徒歩 7 分です。"}, "id": "test_parse"}' \
  http://localhost:5000/api
```

上の例は、ジオパーズ 処理を行う *geonlp.parse* メソッドを呼び出し、パラメータとして送信した文字列を解析した結果を受け取ります。各メソッドで利用できるパラメータやレスポンスの内容については、*WebAPI* **一覧** から対応するメソッドの説明を参照してください。

注釈: WebAPI およびオプションは状態を保持しません。つまり、直前にどのようなリクエストを実行したかに関わらず、同じリクエストを送れば同じレスポンスが返ります。

ただしサーバ側のバージョンやデータベースが変更された場合は、結果が変わることがあります。

7.5 WebAPI 一覧

以下の WebAPI メソッドが利用できます。

7.5.1 geonlp.parse

テキストを geoparse します。

リクエストパラメータ

sentence

[str]

- 変換したいテキスト
- 長さの上限なし

options

[dict, optional]

- *Parse* **オプション** を参照

レスポンス

`features` に GeoJSON Feature 形式の地名語、非地名語、住所をリストとして含む FeatureCollection 形式の GeoJSON を返します。

リクエストの例

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "NII は神保町駅から徒歩 7 分です。"
  },
  "id": "test_parse"
}
```

レスポンスの例


```
{
  "features": [
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "*",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "組織",
          "subclass3": "*",
          "surface": "NII",
          "yomi": ""
        },
        "node_type": "NORMAL",
        "surface": "NII"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "は",
          "pos": "助詞",
          "prononciation": "ワ",
          "subclass1": "係助詞",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "は",
          "yomi": "ハ"
        },
        "node_type": "NORMAL",
```


(次のページに続く)

(前のページからの続き)


```
      "surface": "は"
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [
        139.757845,
        35.6960275
      ],
      "type": "Point"
    },
    "properties": {
      "geoword_properties": {
        "body": "神保町",
        "dictionary_id": 3,
        "dictionary_identifier": "geonlp:ksj-station-N02",
        "entry_id": "5WS6qh",
        "geolod_id": "uN6ecI",
        "hypernym": [
          "東京都",
          "10 号線新宿線"
        ],
        "institution_type": "公営鉄道",
        "latitude": "35.6960275",
        "longitude": "139.757845",
        "ne_class": "鉄道施設/鉄道駅",
        "railway_class": "普通鉄道",
        "suffix": [
          "駅",
          ""
        ]
      },
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "神保町駅",
        "pos": "名詞",
        "prononciation": "",
```


(次のページに続く)

(前のページからの続き)


```
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "uN6ecI:神保町駅",
        "surface": "神保町駅",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "神保町駅"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "から",
        "pos": "助詞",
        "prononciation": "カラ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "から",
        "yomi": "カラ"
      },
      "node_type": "NORMAL",
      "surface": "から"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "徒歩",
        "pos": "名詞",
```


(次のページに続く)

(前のページからの続き)


```
        "prononciation": "トホ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "徒歩",
        "yomi": "トホ"
      },
      "node_type": "NORMAL",
      "surface": "徒歩"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "7",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "7"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "分",
```


(次のページに続く)

(前のページからの続き)


```
        "pos": "名詞",
        "prononciation": "フン",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "分",
        "yomi": "フン"
      },
      "node_type": "NORMAL",
      "surface": "分"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・デス",
        "conjugation_type": "基本形",
        "original_form": "です",
        "pos": "助動詞",
        "prononciation": "デス",
        "subclass1": "*",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "です",
        "yomi": "デス"
      },
      "node_type": "NORMAL",
      "surface": "です"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
```


(次のページに続く)

(前のページからの続き)


```
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```


7.5.2 geonlp.parseStructured

複数のセンテンスを geoparse します。

このメソッドは、リストのそれぞれのテキストを geoparse してから結果を結合します。

先にテキストを結合して `parse()` を呼んだ場合と比較すると、長すぎるテキストの分割を自動で行なうか、呼びだす側で指定するかの違いがあります。

自動的に分割すると、意味的に連続したパラグラフの途中で切れてしまうことがあり、地名解決の精度が低下します。そのためテキストの意味的な区切り（文、段落など）が分かっている場合は、1 ブロックずつ `parse()` で処理するか、`parseStructured()` を利用してください。

リクエストパラメータ

sentence_list

[list of str]

- 変換したいテキストのリスト
- 長さの上限なし、件数の上限なし

options

[dict, optional]

- *Parse* **オプション** を参照

レスポンス

`features` に GeoJSON Feature 形式の地名語、非地名語、住所をリストとして含む FeatureCollection 形式の GeoJSON を返します。

リクエストの例

```
{
  "method": "geonlp.parseStructured",
  "params": {
    "sentence_list": [
      "NII は神保町駅から徒歩 7 分です。",
      "千代田区一ツ橋 2-1-2 にあります。",
      "竹橋駅も近いです。"
    ]
  },
  "id": "test_parseStructured"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "*",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "組織",
          "subclass3": "*",
          "surface": "NII",
          "yomi": ""
        },
```

(次のページに続く)

(前のページからの続き)


```
      "node_type": "NORMAL",
      "surface": "NII"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "は",
        "pos": "助詞",
        "prononciation": "ワ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "は",
        "yomi": "ハ"
      },
      "node_type": "NORMAL",
      "surface": "は"
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [
        139.757845,
        35.6960275
      ],
      "type": "Point"
    },
    "properties": {
      "geoword_properties": {
        "body": "神保町",
        "dictionary_id": 3,
        "dictionary_identifier": "geonlp:ksj-station-N02",
        "entry_id": "5WS6qh",
```


(次のページに続く)

(前のページからの続き)


```
        "geolod_id": "uN6ecI",
        "hypernym": [
          "東京都",
          "10 号線新宿線"
        ],
        "institution_type": "公営鉄道",
        "latitude": "35.6960275",
        "longitude": "139.757845",
        "ne_class": "鉄道施設/鉄道駅",
        "railway_class": "普通鉄道",
        "suffix": [
          "駅",
          ""
        ]
      },
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "神保町駅",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "uN6ecI: 神保町駅",
        "surface": "神保町駅",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "神保町駅"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "から",
```


(次のページに続く)

(前のページからの続き)


```
        "pos": "助詞",
        "prononciation": "カラ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "から",
        "yomi": "カラ"
      },
      "node_type": "NORMAL",
      "surface": "から"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "徒歩",
        "pos": "名詞",
        "prononciation": "トホ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "徒歩",
        "yomi": "トホ"
      },
      "node_type": "NORMAL",
      "surface": "徒歩"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "7",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "7"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "分",
        "pos": "名詞",
        "prononciation": "フン",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "分",
        "yomi": "フン"
      },
      "node_type": "NORMAL",
      "surface": "分"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・デス",
```


(次のページに続く)

(前のページからの続き)


```
        "conjugation_type": "基本形",
        "original_form": "です",
        "pos": "助動詞",
        "prononciation": "デス",
        "subclass1": "*",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "です",
        "yomi": "デス"
      },
      "node_type": "NORMAL",
      "surface": "です"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "。",
        "pos": "記号",
        "prononciation": "。",
        "subclass1": "句点",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "。",
        "yomi": "。"
      },
      "node_type": "NORMAL",
      "surface": "。"
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [
        139.753634,
```


(次のページに続く)

(前のページからの続き)


```
          35.694003
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
          "address": "東京都千代田区",
          "body": "千代田",
          "body_variants": "千代田",
          "code": {},
          "countyname": "",
          "countyname_variants": "",
          "dictionary_id": 1,
          "dictionary_identifier": "geonlp:geoshape-city",
          "entry_id": "13101A1968",
          "geolod_id": "WWIY7G",
          "hypernym": [
            "東京都"
          ],
          "latitude": "35.69400300",
          "longitude": "139.75363400",
          "ne_class": "市区町村",
          "prefname": "東京都",
          "prefname_variants": "東京都",
          "source": "1/千代田区役所/千代田区九段南 1-2-1/P34-14_13.xml",
          "suffix": [
            "区"
          ],
          "valid_from": "",
          "valid_to": ""
        },
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "千代田区",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "地名語",
```


(次のページに続く)

(前のページからの続き)


```
        "subclass3": "WWIY7G:千代田区",
        "surface": "千代田区",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "千代田区"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "一ツ橋",
        "pos": "名詞",
        "prononciation": "ヒトツバシ",
        "subclass1": "固有名詞",
        "subclass2": "地域",
        "subclass3": "一般",
        "surface": "一ツ橋",
        "yomi": "ヒトツバシ"
      },
      "node_type": "NORMAL",
      "surface": "一ツ橋"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
```


(次のページに続く)

(前のページからの続き)


```
          "subclass2": "*",
          "subclass3": "*",
          "surface": "2",
          "yomi": ""
        },
        "node_type": "NORMAL",
        "surface": "2"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "*",
          "pos": "記号",
          "prononciation": "",
          "subclass1": "一般",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "-",
          "yomi": ""
        },
        "node_type": "NORMAL",
        "surface": "-"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "*",
          "pos": "名詞",
          "prononciation": "",
```


(次のページに続く)

(前のページからの続き)


```
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "1",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "1"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "記号",
        "prononciation": "",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "-",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "-"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
```


(次のページに続く)

(前のページからの続き)


```
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "2",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "2"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "に",
        "pos": "助詞",
        "prononciation": "ニ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "に",
        "yomi": "ニ"
      },
      "node_type": "NORMAL",
      "surface": "に"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "五段・ラ行",
        "conjugation_type": "連用形",
        "original_form": "ある",
```


(次のページに続く)

(前のページからの続き)


```
        "pos": "動詞",
        "prononciation": "アリ",
        "subclass1": "自立",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "あり",
        "yomi": "アリ"
      },
      "node_type": "NORMAL",
      "surface": "あり"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・マス",
        "conjugation_type": "基本形",
        "original_form": "ます",
        "pos": "助動詞",
        "prononciation": "マス",
        "subclass1": "*",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "ます",
        "yomi": "マス"
      },
      "node_type": "NORMAL",
      "surface": "ます"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "original_form": "。",
        "pos": "記号",
        "prononciation": "。",
        "subclass1": "句点",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "。",
        "yomi": "。"
      },
      "node_type": "NORMAL",
      "surface": "。"
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [
        139.75767000000002,
        35.69044
      ],
      "type": "Point"
    },
    "properties": {
      "geoword_properties": {
        "body": "竹橋",
        "dictionary_id": 3,
        "dictionary_identifier": "geonlp:ksj-station-N02",
        "entry_id": "976eu3",
        "geolod_id": "3QKgos",
        "hypernym": [
          "東京地下鉄",
          "5 号線東西線"
        ],
        "institution_type": "民営鉄道",
        "latitude": "35.69044",
        "longitude": "139.75767000000002",
        "ne_class": "鉄道施設/鉄道駅",
        "railway_class": "普通鉄道",
        "suffix": [
```


(次のページに続く)

(前のページからの続き)


```
          "駅",
          ""
        ]
      },
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "竹橋駅",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "3QKgos:竹橋駅",
        "surface": "竹橋駅",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "竹橋駅"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "も",
        "pos": "助詞",
        "prononciation": "モ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "も",
        "yomi": "モ"
      },
      "node_type": "NORMAL",
      "surface": "も"
    },
```


(次のページに続く)

(前のページからの続き)


```
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "形容詞・アウオ段",
        "conjugation_type": "基本形",
        "original_form": "近い",
        "pos": "形容詞",
        "prononciation": "チカイ",
        "subclass1": "自立",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "近い",
        "yomi": "チカイ"
      },
      "node_type": "NORMAL",
      "surface": "近い"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・デス",
        "conjugation_type": "基本形",
        "original_form": "です",
        "pos": "助動詞",
        "prononciation": "デス",
        "subclass1": "*",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "です",
        "yomi": "デス"
      },
      "node_type": "NORMAL",
      "surface": "です"
```


(次のページに続く)

(前のページからの続き)


```
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```


7.5.3 geonlp.search

地名語の情報をデータベースから検索します。

リクエストパラメータ

key

[str]

- 検索したい語の表記または読み

options

[dict, optional]

- *Parse* **オプション** を参照

レスポンス

geolod_id をキー、地名語の情報を値に持つ JSON オブジェクトを返します。

リクエストの例

```
{
  "method": "geonlp.search",
  "params": {
    "key": "国会議事堂前"
  },
  "id": "test_search"
}
```

レスポンスの例

```
{
  "Bn4q6d": {
    "body": "国会議事堂前",
    "dictionary_id": 3,
    "dictionary_identifier": "geonlp:ksj-station-N02",
    "entry_id": "LrGGxY",
    "geolod_id": "Bn4q6d",
    "hypernym": [
      "東京地下鉄",
      "4 号線丸ノ内線"
    ],
    "institution_type": "民営鉄道",
    "latitude": "35.674845",
    "longitude": "139.74534166666666",
    "ne_class": "鉄道施設/鉄道駅",
    "railway_class": "普通鉄道",
    "suffix": [
      "駅",
      ""
    ]
  },
  "cE8W4w": {
```

(次のページに続く)

(前のページからの続き)


```
    "body": "国会議事堂前",
    "dictionary_id": 3,
    "dictionary_identifier": "geonlp:ksj-station-N02",
    "entry_id": "4NFELa",
    "geolod_id": "cE8W4w",
    "hypernym": [
      "東京地下鉄",
      "9 号線千代田線"
    ],
    "institution_type": "民営鉄道",
    "latitude": "35.673543333333335",
    "longitude": "139.74305333333334",
    "ne_class": "鉄道施設/鉄道駅",
    "railway_class": "普通鉄道",
    "suffix": [
      "駅",
      ""
    ]
  }
}
```


7.5.4 geonlp.getGeoInfo

指定した geolod_id を持つ語の情報を返します。id がデータベースに存在しない場合は null を返します。

リクエストパラメータ

key

[list of str]

- 検索する語の geolod_id のリスト

options

[dict, optional]

- *Parse* **オプション** を参照

レスポンス

geolod_id をキー、地名語の情報を値に持つ JSON オブジェクトを返します。

リクエストの例

```
{
  "method": "geonlp.getGeoInfo",
  "params": {
    "idlist": [
      "Bn4q6d",
      "cE8W4w"
    ]
  },
  "id": "test_getGeoInfo_idlist"
}
```

レスポンスの例

```
{
  "Bn4q6d": {
    "body": "国会議事堂前",
    "dictionary_id": 3,
    "dictionary_identifier": "geonlp:ksj-station-N02",
    "entry_id": "LrGGxY",
    "geolod_id": "Bn4q6d",
    "hypernym": [
      "東京地下鉄",
      "4 号線丸ノ内線"
    ],
    "institution_type": "民営鉄道",
    "latitude": "35.674845",
    "longitude": "139.74534166666666",
    "ne_class": "鉄道施設/鉄道駅",
    "railway_class": "普通鉄道",
    "suffix": [
      "駅",
      ""
    ]
  },
  "cE8W4w": {
    "body": "国会議事堂前",
    "dictionary_id": 3,
    "dictionary_identifier": "geonlp:ksj-station-N02",
```

(次のページに続く)

(前のページからの続き)

```
    "entry_id": "4NFELa",
    "geolod_id": "cE8W4w",
    "hypernym": [
      "東京地下鉄",
      "9 号線千代田線"
    ],
    "institution_type": "民営鉄道",
    "latitude": "35.673543333333335",
    "longitude": "139.74305333333334",
    "ne_class": "鉄道施設/鉄道駅",
    "railway_class": "普通鉄道",
    "suffix": [
      "駅",
      ""
    ]
  }
}
```

7.5.5 geonlp.getDictionaries

データベースに登録されている地名解析辞書の identifier のリストを返します。

リクエストパラメータ

options
: [dict, optional]

  - *Parse* **オプション** を参照

レスポンス

辞書 identifier のリストを返します。

リクエストの例

```
{
  "method": "geonlp.getDictionaries",
  "params": {},
  "id": "test_getDictionaries"
}
```

レスポンスの例

```
[
  "geonlp:geoshape-city",
  "geonlp:geoshape-pref",
  "geonlp:ksj-station-N02"
]
```

7.5.6 geonlp.getDictionaryInfo

指定された identifer を持つ地名解析辞書の JSONLD メタデータをデータベースから取得します。

結果は JSONLD 文字列です。JSONLD をデコードした JSON オブジェクトではない点に注意してください。

リクエストパラメータ

identifier

[str]

- 地名解析辞書の identifier

options

[dict, optional]

- *Parse* **オプション** を参照

レスポンス

辞書の jsonld メタデータ文字列を返します。identifier と一致する辞書が存在しない場合には null を返します。

リクエストの例

```
{
  "method": "geonlp.getDictionaryInfo",
  "params": {
    "identifier": "geonlp:ksj-station-N02"
  },
  "id": "test_getDictionaryInfo"
}
```

レスポンスの例


```
"{\"@context\": \"https://schema.org/\", \"@type\": \"Dataset\", \"alternateName\": \"\",
 \"creator\": [{\"@type\": \"Organization\", \"name\": \"株式会社情報試作室\", \"sameAs\":␣
↪\"https://www.info-proto.com/\"}], \"dateModified\": \"2021-08-27T17:18:18+09:00\", \
↪"description\": \"国土数値情報「鉄道データ（N02)」から作成した、日本の鉄道駅（地下鉄を含む）の辞書
です。hypernym には運営者名と路線名を記載しています。「都営」ではなく「東京都」のようになっていますの
で注意してください。自由フィールドとして、railway_class に「鉄道区分」、institution_type に「事業者
種別」を含みます。\", \"distribution\": [{\"@type\": \"DataDownload\", \"contentUrl\": \
↪"https://www.info-proto.com/static/ksj-station-N02.csv\", \"encodingFormat\": \"text/
↪csv\"}], \"identifier\": [\"geonlp:ksj-station-N02\"], \"isBasedOn\": {\"@type\": \
↪"CreativeWork\", \"name\": \"国土数値情報 鉄道データ\", \"url\": \"https://nlftp.mlit.go.
↪jp/ksj/gml/datalist/KsjTmplt-N02-v2_2.html\"}, \"keywords\": [\"GeoNLP\", \"地名辞書\"],
 \"license\": \"https://creativecommons.org/licenses/by/4.0/\", \"name\": \"国土数値情報 鉄
道データ（駅）\", \"size\": \"10191\", \"spatialCoverage\": {\"@type\": \"Place\", \"geo\
↪": {\"@type\": \"GeoShape\", \"box\": \"26.193265 127.652285 45.41616333333333 145.
↪59723\"}}, \"temporalCoverage\": \"../..\", \"url\": \"https://www.info-proto.com/
↪static/ksj-station-N02.html\"}"
```


7.5.7 geonlp.addressGeocoding

住所ジオコーディング処理を行ないます。

リクエストパラメータ

address

[str]

- 住所文字列

レスポンス

住所ジオコーディングの結果を JSON で返します。

リクエストの例


```
{
  "method": "geonlp.addressGeocoding",
  "params": {
    "address": "千代田区一ツ橋 2-1-2"
  },
```


(次のページに続く)

(前のページからの続き)

```
  "id": "test_addressGeocoding"
}
```

レスポンスの例

```
{
  "candidates": [
    {
      "fullname": [
        "東京都",
        "千代田区",
        "一ツ橋",
        "二丁目",
        "1 番"
      ],
      "id": 49244643,
      "level": 7,
      "name": "1 番",
      "note": "",
      "priority": 3,
      "x": 139.75814819335938,
      "y": 35.69233322143555
    }
  ],
  "matched": "千代田区一ツ橋 2-1-"
}
```

7.5.8 geonlp.version

Pygeonlp のバージョンを返します。

リクエストパラメータ

なし

レスポンス

バージョン番号（文字列）

リクエストの例

```
{
  "method": "geonlp.version",
  "params": {},
  "id": "test_version"
}
```

レスポンスの例

```
"1.2.2"
```

7.6 Parse オプション

geonlp.parse および *geonlp.parseStructured* はジオパーズ処理を細かく制御するためのオプションを指定することができます。たとえば利用する地名語の地名解析辞書や固有名クラスを制限したり、結果に対して適用するフィルタを指定できます。

以下の Parse オプションが利用できます。

7.6.1 geocoding オプション

住所ジオコーディングを利用するかどうかを指定します。

geonlp.parse, *geonlp.parseStructured* を実行する場合に、このオプションに true がセットされているとテキスト中の住所文字列を住所として解析します。

パラメータ

geocoding
: [bool, optional]

  - true の場合、住所ジオコーディングを行ないます
  - false または省略されると住所ジオコーディングは行ないません

リクエストの例

住所ジオコーディングを利用します。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "NII は千代田区一ツ橋 2-1-2 にあります。",
    "options": {
      "geocoding": true
    }
  },
  "id": "test_parse_geocoding"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "*",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "組織",
          "subclass3": "*",
          "surface": "NII",
          "yomi": ""
        },
        "node_type": "NORMAL",
        "surface": "NII"
      },
      "type": "Feature"
    },
    {
```

(次のページに続く)

(前のページからの続き)


```
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "は",
        "pos": "助詞",
        "prononciation": "ワ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "は",
        "yomi": "ハ"
      },
      "node_type": "NORMAL",
      "surface": "は"
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [
        139.75814819335938,
        35.69233322143555
      ],
      "type": "Point"
    },
    "properties": {
      "address_properties": {
        "fullname": [
          "東京都",
          "千代田区",
          "一ツ橋",
          "二丁目",
          "1 番"
        ],
        "id": 49244643,
        "level": 7,
        "name": "1 番",
```


(次のページに続く)

(前のページからの続き)


```
      "note": "",
      "priority": 3,
      "x": 139.75814819335938,
      "y": 35.69233322143555
    },
    "morphemes": [
      {
        "geometry": {
          "coordinates": [
            139.753634,
            35.694003
          ],
          "type": "Point"
        },
        "properties": {
          "geoword_properties": {
            "address": "東京都千代田区",
            "body": "千代田",
            "body_variants": "千代田",
            "code": {},
            "countyname": "",
            "countyname_variants": "",
            "dictionary_id": 1,
            "dictionary_identifier": "geonlp:geoshape-city",
            "entry_id": "13101A1968",
            "geolod_id": "WWIY7G",
            "hypernym": [
              "東京都"
            ],
            "latitude": "35.69400300",
            "longitude": "139.75363400",
            "ne_class": "市区町村",
            "prefname": "東京都",
            "prefname_variants": "東京都",
            "source": "1/千代田区役所/千代田区九段南 1-2-1/P34-14_13.xml",
            "suffix": [
              "区"
            ],
            "valid_from": "",
```


(次のページに続く)

(前のページからの続き)


```
        "valid_to": ""
      },
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "千代田区",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "WWIY7G:千代田区",
        "surface": "千代田区",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "千代田区"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "一ツ橋",
        "pos": "名詞",
        "prononciation": "ヒトツバシ",
        "subclass1": "固有名詞",
        "subclass2": "地域",
        "subclass3": "一般",
        "surface": "一ツ橋",
        "yomi": "ヒトツバシ"
      },
      "node_type": "NORMAL",
      "surface": "一ツ橋"
    },
    "type": "Feature"
  },
```


(次のページに続く)

(前のページからの続き)


```
        {
          "geometry": null,
          "properties": {
            "morphemes": {
              "conjugated_form": "*",
              "conjugation_type": "*",
              "original_form": "*",
              "pos": "名詞",
              "prononciation": "",
              "subclass1": "数",
              "subclass2": "*",
              "subclass3": "*",
              "surface": "2",
              "yomi": ""
            },
            "node_type": "NORMAL",
            "surface": "2"
          },
          "type": "Feature"
        },
        {
          "geometry": null,
          "properties": {
            "morphemes": {
              "conjugated_form": "*",
              "conjugation_type": "*",
              "original_form": "*",
              "pos": "記号",
              "prononciation": "",
              "subclass1": "一般",
              "subclass2": "*",
              "subclass3": "*",
              "surface": "-",
              "yomi": ""
            },
            "node_type": "NORMAL",
            "surface": "-"
          },
          "type": "Feature"
```


(次のページに続く)

(前のページからの続き)


```
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "*",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "数",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "1",
          "yomi": ""
        },
        "node_type": "NORMAL",
        "surface": "1"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "*",
          "pos": "記号",
          "prononciation": "",
          "subclass1": "一般",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "-",
          "yomi": ""
        },
        "node_type": "NORMAL",
        "surface": "-"
      },
```


(次のページに続く)

(前のページからの続き)

```
            "type": "Feature"
          }
        ],
        "node_type": "ADDRESS",
        "surface": "千代田区一ツ橋 2-1-"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "*",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "数",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "2",
          "yomi": ""
        },
        "node_type": "NORMAL",
        "surface": "2"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "に",
          "pos": "助詞",
          "prononciation": "ニ",
          "subclass1": "格助詞",
          "subclass2": "一般",
```

(次のページに続く)

(前のページからの続き)


```
        "subclass3": "*",
        "surface": "に",
        "yomi": "ニ"
      },
      "node_type": "NORMAL",
      "surface": "に"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "五段・ラ行",
        "conjugation_type": "連用形",
        "original_form": "ある",
        "pos": "動詞",
        "prononciation": "アリ",
        "subclass1": "自立",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "あり",
        "yomi": "アリ"
      },
      "node_type": "NORMAL",
      "surface": "あり"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・マス",
        "conjugation_type": "基本形",
        "original_form": "ます",
        "pos": "助動詞",
        "prononciation": "マス",
        "subclass1": "*",
```


(次のページに続く)

(前のページからの続き)


```
          "subclass2": "*",
          "subclass3": "*",
          "surface": "ます",
          "yomi": "マス"
        },
        "node_type": "NORMAL",
        "surface": "ます"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```

7.6.2 set-dic オプション

利用する地名解析辞書を指定します。

指定しない場合はデータベースに登録されている全ての地名解析辞書が利用されます。

パラメータ

set-dic : str, list of str

- 文字列が指定された場合、正規表現として解釈し、**パターンを含む** identifier を持つ辞書を利用します
- 文字列のリストが指定された場合、利用する辞書 identifier のリストと解釈し、リストに含まれる identifier 持つ辞書を利用します

リクエストの例

identifier に 'geoshape' を含む辞書 (geonlp:geoshape-city, geonlp:geoshape-pref) を利用するように指定します。駅名は検索されなくなります。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "和歌山市は晴れ。",
    "options": {
      "set-dic": "geoshape"
    }
  },
  "id": "test_parse_set_dic"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          135.170808,
          34.230514
        ],
        "type": "Point"
```

(次のページに続く)

(前のページからの続き)


```
    },
    "properties": {
      "geoword_properties": {
        "address": "和歌山県和歌山市",
        "body": "和歌山",
        "body_variants": "和歌山",
        "code": {},
        "countyname": "",
        "countyname_variants": "",
        "dictionary_id": 1,
        "dictionary_identifier": "geonlp:geoshape-city",
        "entry_id": "30201A1968",
        "geolod_id": "lQccqK",
        "hypernym": [
          "和歌山県"
        ],
        "latitude": "34.23051400",
        "longitude": "135.17080800",
        "ne_class": "市区町村",
        "prefname": "和歌山県",
        "prefname_variants": "和歌山県",
        "source": "1/和歌山市役所/和歌山市七番丁 23/P34-14_30.xml",
        "suffix": [
          "市"
        ],
        "valid_from": "1889-04-01",
        "valid_to": ""
      },
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "和歌山市",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "lQccqK:和歌山市",
        "surface": "和歌山市",
        "yomi": ""
```


(次のページに続く)

(前のページからの続き)


```
      },
      "node_type": "GEOWORD",
      "surface": "和歌山市"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "は",
        "pos": "助詞",
        "prononciation": "ワ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "は",
        "yomi": "ハ"
      },
      "node_type": "NORMAL",
      "surface": "は"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "晴れ",
        "pos": "名詞",
        "prononciation": "ハレ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "晴れ",
```


(次のページに続く)

(前のページからの続き)


```
          "yomi": "ハレ"
        },
        "node_type": "NORMAL",
        "surface": "晴れ"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```


7.6.3 remove-dic オプション

利用しない地名解析辞書を指定します。

set-dic **オプション** と併用した場合、まず `set-dic` を評価して利用する辞書を決定し、そこから `remove-dic` で指定された辞書を除外します。

`set-dic` が指定されていない場合、データベースに登録されている全ての辞書から `remove-dic` で指定された辞

書を除外します。

パラメータ

remove-dic : str, list of str

- 文字列が指定された場合、正規表現として解釈し、**パターンを含む** identifier を持つ辞書を除外します
- 文字列のリストが指定された場合、利用する辞書 identifier のリストと解釈し、リストに含まれる identifier 持つ辞書を除外します

リクエストの例

identifier に 'station' を含む辞書 (geonlp:ksj-station-N02-2019) を利用しないように指定します。駅名は検索されなくなります。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "和歌山市は晴れ。",
    "options": {
      "remove-dic": "station"
    }
  },
  "id": "test_parse_remove_dic"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          135.170808,
          34.230514
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
```

(次のページに続く)

(前のページからの続き)


```
        "address": "和歌山県和歌山市",
        "body": "和歌山",
        "body_variants": "和歌山",
        "code": {},
        "countyname": "",
        "countyname_variants": "",
        "dictionary_id": 1,
        "dictionary_identifier": "geonlp:geoshape-city",
        "entry_id": "30201A1968",
        "geolod_id": "lQccqK",
        "hypernym": [
          "和歌山県"
        ],
        "latitude": "34.23051400",
        "longitude": "135.17080800",
        "ne_class": "市区町村",
        "prefname": "和歌山県",
        "prefname_variants": "和歌山県",
        "source": "1/和歌山市役所/和歌山市七番丁 23/P34-14_30.xml",
        "suffix": [
          "市"
        ],
        "valid_from": "1889-04-01",
        "valid_to": ""
      },
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "和歌山市",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "lQccqK:和歌山市",
        "surface": "和歌山市",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "和歌山市"
```


(次のページに続く)

(前のページからの続き)


```
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "は",
        "pos": "助詞",
        "prononciation": "ワ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "は",
        "yomi": "ハ"
      },
      "node_type": "NORMAL",
      "surface": "は"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "晴れ",
        "pos": "名詞",
        "prononciation": "ハレ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "晴れ",
        "yomi": "ハレ"
      },
      "node_type": "NORMAL",
```


(次のページに続く)

(前のページからの続き)

```
        "surface": "晴れ"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```

7.6.4 add-dic オプション

利用する地名解析辞書を追加します。

set-dic **オプション** や *remove-dic* **オプション** と併用した場合、まず set-dic と remove-dic を評価して利用する辞書を決定し、そこに add-dic で指定された辞書を追加します。

set-dic も remove-dic も指定されていない場合、データベースに登録されている全ての辞書を利用するので、add-dic の指定は意味がありません。

パラメータ

add-dic : str, list of str

- 文字列が指定された場合、正規表現として解釈し、**パターンを含む** identifier を持つ辞書を利用対象に追加します
- 文字列のリストが指定された場合、利用する辞書 identifier のリストと解釈し、リストに含まれる identifier 持つ辞書を利用対象に追加します

リクエストの例

まず `remove-dic` で全ての辞書を除外し、次に `add-dic` で identifier に 'city' を含む辞書 (geonlp:geoshape-city) を利用する辞書として追加します。市区町村名しか検索されなくなります。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "和歌山市は晴れ。",
    "options": {
      "remove-dic": ".*",
      "add-dic": "city"
    }
  },
  "id": "test_parse_add_dic"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          135.170808,
          34.230514
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
          "address": "和歌山県和歌山市",
          "body": "和歌山",
```

(次のページに続く)

(前のページからの続き)


```
        "body_variants": "和歌山",
        "code": {},
        "countyname": "",
        "countyname_variants": "",
        "dictionary_id": 1,
        "dictionary_identifier": "geonlp:geoshape-city",
        "entry_id": "30201A1968",
        "geolod_id": "lQcccqK",
        "hypernym": [
          "和歌山県"
        ],
        "latitude": "34.23051400",
        "longitude": "135.17080800",
        "ne_class": "市区町村",
        "prefname": "和歌山県",
        "prefname_variants": "和歌山県",
        "source": "1/和歌山市役所/和歌山市七番丁 23/P34-14_30.xml",
        "suffix": [
          "市"
        ],
        "valid_from": "1889-04-01",
        "valid_to": ""
      },
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "和歌山市",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "lQcccqK:和歌山市",
        "surface": "和歌山市",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "和歌山市"
    },
    "type": "Feature"
```


(次のページに続く)

(前のページからの続き)


```
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "は",
          "pos": "助詞",
          "prononciation": "ワ",
          "subclass1": "係助詞",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "は",
          "yomi": "ハ"
        },
        "node_type": "NORMAL",
        "surface": "は"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "晴れ",
          "pos": "名詞",
          "prononciation": "ハレ",
          "subclass1": "一般",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "晴れ",
          "yomi": "ハレ"
        },
        "node_type": "NORMAL",
        "surface": "晴れ"
      },
```


(次のページに続く)

(前のページからの続き)


```
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```


7.6.5 set-class オプション

解析対象とする固有名クラスを指定します。

固有名クラスは正規表現文字列のリストで指定してください。また、除外したいクラスは先頭に - を指定してください。

set-class **オプション**, *remove-class* **オプション**, *add-class* **オプション**が同時に指定されている場合、まず `set-class`, 次に `remove-class`, 最後に `add-class` が評価されます。

`set-class` を指定しない場合、対象クラスは `[r'.*']` になります。つまりデータベース内の全ての地名語が対象です。

クラス名の評価は指定された順番に行なわれます。つまり「鉄道施設は除く、ただし駅は利用する」という場合は

`[r'.*', r'-鉄道施設/.*', r'.*駅$']` のように指定します。

パラメータ

set-class : list of str

- 正規表現として解釈し、**パターンに一致する**固有名クラスを持つ地名語を解析対象とします

リクエストの例

全ての固有名クラスから **鉄道施設/.*** に一致するものを除外したクラスを検索対象とします。駅名は検索されなくなります。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "和歌山市は晴れ。",
    "options": {
      "set-class": [
        ".*",
        "-鉄道施設/.*"
      ]
    }
  },
  "id": "test_parse_set_class"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          135.170808,
          34.230514
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
```

(次のページに続く)

(前のページからの続き)


```
      "address": "和歌山県和歌山市",
      "body": "和歌山",
      "body_variants": "和歌山",
      "code": {},
      "countyname": "",
      "countyname_variants": "",
      "dictionary_id": 1,
      "dictionary_identifier": "geonlp:geoshape-city",
      "entry_id": "30201A1968",
      "geolod_id": "lQcccqK",
      "hypernym": [
        "和歌山県"
      ],
      "latitude": "34.23051400",
      "longitude": "135.17080800",
      "ne_class": "市区町村",
      "prefname": "和歌山県",
      "prefname_variants": "和歌山県",
      "source": "1/和歌山市役所/和歌山市七番丁 23/P34-14_30.xml",
      "suffix": [
        "市"
      ],
      "valid_from": "1889-04-01",
      "valid_to": ""
    },
    "morphemes": {
      "conjugated_form": "*",
      "conjugation_type": "*",
      "original_form": "和歌山市",
      "pos": "名詞",
      "prononciation": "",
      "subclass1": "固有名詞",
      "subclass2": "地名語",
      "subclass3": "lQcccqK:和歌山市",
      "surface": "和歌山市",
      "yomi": ""
    },
    "node_type": "GEOWORD",
    "surface": "和歌山市"
```


(次のページに続く)

(前のページからの続き)


```
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "は",
        "pos": "助詞",
        "prononciation": "ワ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "は",
        "yomi": "ハ"
      },
      "node_type": "NORMAL",
      "surface": "は"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "晴れ",
        "pos": "名詞",
        "prononciation": "ハレ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "晴れ",
        "yomi": "ハレ"
      },
      "node_type": "NORMAL",
```


(次のページに続く)

(前のページからの続き)


```
        "surface": "晴れ"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```


7.6.6 remove-class オプション

解析対象とする固有名クラスを除外します。

固有名クラスは正規表現文字列のリストで指定してください。また、クラス名の先頭に - を指定すると、そのクラスは除外ではなく追加されます。

set-class **オプション**, *remove-class* **オプション**, *add-class* **オプション**が同時に指定されている場合、まず `set-class`, 次に `remove-class`, 最後に `add-class` が評価されます。

`set-class` が指定されていない場合は `[r'.*']` から除外します。

クラス名の評価は指定された順番に行なわれます。つまり「鉄道施設は除く、ただし鉄道駅は利用する」という場合は `[r'鉄道施設/.*', r'-.*駅$']` のように指定します。

パラメータ

add-class : list of str

- 正規表現として解釈し、**パターンに一致する**固有名クラスを持つ地名語を解析対象に追加します

リクエストの例

まず全ての固有名クラスを検索対象から除外し、次に **市区町村** に一致するものを検索対象に追加します。市区町村名しか検索されなくなります。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "和歌山市は晴れ。",
    "options": {
      "remove-class": [
        ".*",
        "-市区町村"
      ]
    }
  },
  "id": "test_parse_remove_class"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          135.170808,
          34.230514
        ],
        "type": "Point"
      },
      "properties": {
```

(次のページに続く)

(前のページからの続き)


```
      "geoword_properties": {
        "address": "和歌山県和歌山市",
        "body": "和歌山",
        "body_variants": "和歌山",
        "code": {},
        "countyname": "",
        "countyname_variants": "",
        "dictionary_id": 1,
        "dictionary_identifier": "geonlp:geoshape-city",
        "entry_id": "30201A1968",
        "geolod_id": "lQccqK",
        "hypernym": [
          "和歌山県"
        ],
        "latitude": "34.23051400",
        "longitude": "135.17080800",
        "ne_class": "市区町村",
        "prefname": "和歌山県",
        "prefname_variants": "和歌山県",
        "source": "1/和歌山市役所/和歌山市七番丁 23/P34-14_30.xml",
        "suffix": [
          "市"
        ],
        "valid_from": "1889-04-01",
        "valid_to": ""
      },
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "和歌山市",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "lQccqK:和歌山市",
        "surface": "和歌山市",
        "yomi": ""
      },
      "node_type": "GEOWORD",
```


(次のページに続く)

(前のページからの続き)


```
      "surface": "和歌山市"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "は",
        "pos": "助詞",
        "prononciation": "ワ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "は",
        "yomi": "ハ"
      },
      "node_type": "NORMAL",
      "surface": "は"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "晴れ",
        "pos": "名詞",
        "prononciation": "ハレ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "晴れ",
        "yomi": "ハレ"
      },
```


(次のページに続く)

(前のページからの続き)


```
        "node_type": "NORMAL",
        "surface": "晴れ"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```


7.6.7 add-class オプション

解析対象とする固有名クラスを追加します。

固有名クラスは正規表現文字列のリストで指定してください。また、除外したいクラスは先頭に - を指定してください。

set-class **オプション**, *remove-class* **オプション**, *add-class* **オプション**が同時に指定されている場合、まず `set-class`, 次に `remove-class`, 最後に `add-class` が評価されます。

`set-class`, `remove-class` が指定されていない場合は `[r'.*']` に対して追加します。

クラス名の評価は指定された順番に行なわれます。つまり「鉄道施設は除く、ただし鉄道駅は利用する」という場合は `[r'-鉄道施設/.*', r'.*駅$']` のように指定します。

パラメータ

add-class : list of str

- 正規表現として解釈し、**パターンに一致する**固有名クラスを持つ地名語を解析対象に追加します

リクエストの例

まず全ての固有名クラスを検索対象から除外し、次に **市区町村** に一致するものを検索対象に追加します。市区町村名しか検索されなくなります。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "和歌山市は晴れ。",
    "options": {
      "add-class": [
        "-.*",
        "市区町村"
      ]
    }
  },
  "id": "test_parse_add_class"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          135.170808,
          34.230514
        ],
        "type": "Point"
      },
      "properties": {
```

(次のページに続く)

(前のページからの続き)


```
      "geoword_properties": {
        "address": "和歌山県和歌山市",
        "body": "和歌山",
        "body_variants": "和歌山",
        "code": {},
        "countyname": "",
        "countyname_variants": "",
        "dictionary_id": 1,
        "dictionary_identifier": "geonlp:geoshape-city",
        "entry_id": "30201A1968",
        "geolod_id": "lQccqK",
        "hypernym": [
          "和歌山県"
        ],
        "latitude": "34.23051400",
        "longitude": "135.17080800",
        "ne_class": "市区町村",
        "prefname": "和歌山県",
        "prefname_variants": "和歌山県",
        "source": "1/和歌山市役所/和歌山市七番丁 23/P34-14_30.xml",
        "suffix": [
          "市"
        ],
        "valid_from": "1889-04-01",
        "valid_to": ""
      },
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "和歌山市",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "lQccqK:和歌山市",
        "surface": "和歌山市",
        "yomi": ""
      },
      "node_type": "GEOWORD",
```


(次のページに続く)

(前のページからの続き)


```
      "surface": "和歌山市"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "は",
        "pos": "助詞",
        "prononciation": "ワ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "は",
        "yomi": "ハ"
      },
      "node_type": "NORMAL",
      "surface": "は"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "晴れ",
        "pos": "名詞",
        "prononciation": "ハレ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "晴れ",
        "yomi": "ハレ"
      },
```


(次のページに続く)

(前のページからの続き)


```
        "node_type": "NORMAL",
        "surface": "晴れ"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```


7.6.8 geo-contains オプション

`GeoContainsFilter` を作成し、 *geonlp.parse* および *geonlp.parseStructured* の結果に対してこのフィルタを適用します。

パラメータで指定された GeoJSON が表す領域に含まれる地名語以外は地名語ではない固有名詞に置き換えられます。

パラメータ

geo-contains : str, dict

- str の場合、まず GeoJSON 文字列として解釈します。JSON 文字列では無い場合は GeoJSON を返す URL と解釈し、HTTP で問い合わせを行ないます。
- dict の場合、 GeoJSON をデコードしたオブジェクトと解釈します。 FeatureCollection, Feature, および geometry に対応します。

リクエストの例

東京周辺を含むポリゴンを空間領域として指定し、その領域に含まれる地名語だけを抽出します。「府中」は東京都の府中駅に解決されます。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "府中に行きます。",
    "options": {
      "geo-contains": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              139.43,
              35.54
            ],
            [
              139.91,
              35.54
            ],
            [
              139.91,
              35.83
            ],
            [
              139.43,
              35.83
            ],
            [
              139.43,
              35.54
            ]
          ]
```

(次のページに続く)

(前のページからの続き)

```
        ]
      }
    }
  },
  "id": "test_parse_geo_contains"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          139.4801,
          35.67219
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
          "body": "府中",
          "dictionary_id": 3,
          "dictionary_identifier": "geonlp:ksj-station-N02",
          "entry_id": "KajwsU",
          "geolod_id": "8NE02H",
          "hypernym": [
            "京王電鉄",
            "京王線"
          ],
          "institution_type": "民営鉄道",
          "latitude": "35.67219",
          "longitude": "139.4801",
          "ne_class": "鉄道施設/鉄道駅",
          "railway_class": "普通鉄道",
          "suffix": [
            "駅",
            ""
```

(次のページに続く)

(前のページからの続き)


```
        ]
      },
      "morphemes": {
        "conjugated_form": "",
        "conjugation_type": "*",
        "original_form": "府中",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "8NE02H:府中駅",
        "surface": "府中",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "府中"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "に",
        "pos": "助詞",
        "prononciation": "ニ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "に",
        "yomi": "ニ"
      },
      "node_type": "NORMAL",
      "surface": "に"
    },
    "type": "Feature"
  },
```


(次のページに続く)

(前のページからの続き)


```
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "五段・カ行促音便",
        "conjugation_type": "連用形",
        "original_form": "行く",
        "pos": "動詞",
        "prononciation": "イキ",
        "subclass1": "自立",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "行き",
        "yomi": "イキ"
      },
      "node_type": "NORMAL",
      "surface": "行き"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・マス",
        "conjugation_type": "基本形",
        "original_form": "ます",
        "pos": "助動詞",
        "prononciation": "マス",
        "subclass1": "*",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "ます",
        "yomi": "マス"
      },
      "node_type": "NORMAL",
      "surface": "ます"
    },
    "type": "Feature"
```


(次のページに続く)

(前のページからの続き)


```
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```


7.6.9 geo-disjoint オプション

`GeoDisjointFilter` を作成し、 *geonlp.parse* および *geonlp.parseStructured* の結果に対してこのフィルタを適用します。

パラメータで指定された GeoJSON が表す領域に含まれない地名語以外は地名語ではない固有名詞に置き換えられます。

パラメータ

geo-disjoint : str, dict

- str の場合、まず GeoJSON 文字列として解釈します。JSON 文字列では無い場合は GeoJSON を返す URL と解釈し、HTTP で問い合わせを行ないます。
- dict の場合、 GeoJSON をデコードしたオブジェクトと解釈します。 FeatureCollection, Feature, および

geometry に対応します。

リクエストの例

東京周辺を含むポリゴンを空間領域として指定し、その領域に含まれない地名語だけを抽出します。「府中」は京都府の府中駅に解決されます。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "府中に行きます。",
    "options": {
      "geo-disjoint": "https://geoshape.ex.nii.ac.jp/city/geojson/20200101/13/13208A1968.
↪geojson"
    }
  },
  "id": "test_parse_geo_disjoint"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          135.195275,
          35.583365
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
          "body": "府中",
          "dictionary_id": 3,
          "dictionary_identifier": "geonlp:ksj-station-N02",
          "entry_id": "MREv4E",
          "geolod_id": "We02Nd",
          "hypernym": [
            "丹後海陸交通",
```

(次のページに続く)

(前のページからの続き)


```
            "天橋立鋼索鉄道"
          ],
          "institution_type": "民営鉄道",
          "latitude": "35.583365",
          "longitude": "135.195275",
          "ne_class": "鉄道施設/鉄道駅",
          "railway_class": "鋼索鉄道",
          "suffix": [
            "駅",
            ""
          ]
        },
        "morphemes": {
          "conjugated_form": "",
          "conjugation_type": "*",
          "original_form": "府中",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "地名語",
          "subclass3": "We02Nd:府中駅",
          "surface": "府中",
          "yomi": ""
        },
        "node_type": "GEOWORD",
        "surface": "府中"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "に",
          "pos": "助詞",
          "prononciation": "ニ",
          "subclass1": "格助詞",
```


(次のページに続く)

(前のページからの続き)


```
          "subclass2": "一般",
          "subclass3": "*",
          "surface": "に",
          "yomi": "ニ"
        },
        "node_type": "NORMAL",
        "surface": "に"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "五段・カ行促音便",
          "conjugation_type": "連用形",
          "original_form": "行く",
          "pos": "動詞",
          "prononciation": "イキ",
          "subclass1": "自立",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "行き",
          "yomi": "イキ"
        },
        "node_type": "NORMAL",
        "surface": "行き"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "特殊・マス",
          "conjugation_type": "基本形",
          "original_form": "ます",
          "pos": "助動詞",
          "prononciation": "マス",
```


(次のページに続く)

(前のページからの続き)


```
          "subclass1": "*",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "ます",
          "yomi": "マス"
        },
        "node_type": "NORMAL",
        "surface": "ます"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```

7.6.10 time-exists オプション

`TimeExistsFilter` を作成し、 *geonlp.parse* および *geonlp.parseStructured* の結果に対してこのフィルタを適用します。

パラメータで指定された時点または期間に存在した地名語以外は地名語ではない固有名詞に置き換えられます。

パラメータ

time-exists : str, list of str, dict

- str の場合、一時点を表す日付または日時と解釈します。
- list の場合、要素が 1 つならば一時点を、2 つならば期間を表す日付または日時と解釈します。
- dict の場合、 `date_from` の値が期間の開始時点を、`date_to` の値が期間の終了時点を表す日付または日時と解釈します。 `date_to` は省略可能です。

リクエストの例

2000 年 1 月 1 日から 2001 年 2 月 1 日の間に存在した地名語だけを抽出します。「田無市」「保谷市」はこの期間内に合併して「西東京市」になったので、「田無市」「保谷市」「西東京市」は地名語として解釈されます。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "田無市と保谷市は 2001 年 1 月 21 日に合併して西東京市になりました。",
    "options": {
      "time-exists": [
        "2000-01-01",
        "2001-02-01"
      ]
    }
  },
  "id": "test_time_exists"
}
```

レスポンスの例


```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          139.538159,
          35.725499
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
          "address": "東京都田無市",
          "body": "田無",
          "body_variants": "田無",
          "code": {},
          "countyname": "",
          "countyname_variants": "",
          "dictionary_id": 1,
          "dictionary_identifier": "geonlp:geoshape-city",
          "entry_id": "13216A1968",
          "geolod_id": "NGQsEB",
          "hypernym": [
            "東京都"
          ],
          "latitude": "35.72549900",
          "longitude": "139.53815900",
          "ne_class": "市区町村",
          "prefname": "東京都",
          "prefname_variants": "東京都",
          "source": "1/西東京市役所/西東京市南町 5-6-13/P34-14_13.xml/(20001001\\13\\
↪13216A1968.wkt.txt)",
          "suffix": [
            "市"
          ],
          "valid_from": "",
          "valid_to": "2001-01-21"
        },
```


(次のページに続く)

(前のページからの続き)


```
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "田無市",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "地名語",
          "subclass3": "NGQsEB:田無市",
          "surface": "田無市",
          "yomi": ""
        },
        "node_type": "GEOWORD",
        "surface": "田無市"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "と",
          "pos": "助詞",
          "prononciation": "ト",
          "subclass1": "並立助詞",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "と",
          "yomi": "ト"
        },
        "node_type": "NORMAL",
        "surface": "と"
      },
      "type": "Feature"
    },
    {
      "geometry": {
```


(次のページに続く)

(前のページからの続き)


```
        "coordinates": [
          139.558901,
          35.741527
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
          "address": "東京都保谷市",
          "body": "保谷",
          "body_variants": "保谷",
          "code": {},
          "countyname": "",
          "countyname_variants": "",
          "dictionary_id": 1,
          "dictionary_identifier": "geonlp:geoshape-city",
          "entry_id": "13217A1968",
          "geolod_id": "PuWcAj",
          "hypernym": [
            "東京都"
          ],
          "latitude": "35.74152700",
          "longitude": "139.55890100",
          "ne_class": "市区町村",
          "prefname": "東京都",
          "prefname_variants": "東京都",
          "source": "2/保谷庁舎/西東京市中町 1-5-1/P34-14_13.xml/(20001001\\13\\13217A1968.
↪wkt.txt)",
          "suffix": [
            "市"
          ],
          "valid_from": "",
          "valid_to": "2001-01-21"
        },
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "保谷市",
          "pos": "名詞",
```


(次のページに続く)

(前のページからの続き)


```
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "PuWcAj:保谷市",
        "surface": "保谷市",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "保谷市"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "は",
        "pos": "助詞",
        "prononciation": "ワ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "は",
        "yomi": "ハ"
      },
      "node_type": "NORMAL",
      "surface": "は"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "2001",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "2001"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "年",
        "pos": "名詞",
        "prononciation": "ネン",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "年",
        "yomi": "ネン"
      },
      "node_type": "NORMAL",
      "surface": "年"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "1",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "1"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "月",
        "pos": "名詞",
        "prononciation": "ツキ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "月",
        "yomi": "ツキ"
      },
      "node_type": "NORMAL",
      "surface": "月"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "21",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "21"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "日",
        "pos": "名詞",
        "prononciation": "ニチ",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "日",
        "yomi": "ニチ"
      },
      "node_type": "NORMAL",
      "surface": "日"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
```


(次のページに続く)

(前のページからの続き)


```
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "に",
        "pos": "助詞",
        "prononciation": "ニ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "に",
        "yomi": "ニ"
      },
      "node_type": "NORMAL",
      "surface": "に"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "合併",
        "pos": "名詞",
        "prononciation": "ガッペイ",
        "subclass1": "サ変接続",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "合併",
        "yomi": "ガッペイ"
      },
      "node_type": "NORMAL",
      "surface": "合併"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
```


(次のページに続く)

(前のページからの続き)


```
      "morphemes": {
        "conjugated_form": "サ変・スル",
        "conjugation_type": "連用形",
        "original_form": "する",
        "pos": "動詞",
        "prononciation": "シ",
        "subclass1": "自立",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "し",
        "yomi": "シ"
      },
      "node_type": "NORMAL",
      "surface": "し"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "て",
        "pos": "助詞",
        "prononciation": "テ",
        "subclass1": "接続助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "て",
        "yomi": "テ"
      },
      "node_type": "NORMAL",
      "surface": "て"
    },
    "type": "Feature"
  },
  {
    "geometry": {
```


(次のページに続く)

(前のページからの続き)


```
        "coordinates": [
          139.538159,
          35.725499
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
          "address": "東京都西東京市",
          "body": "西東京",
          "body_variants": "西東京",
          "code": {},
          "countyname": "",
          "countyname_variants": "",
          "dictionary_id": 1,
          "dictionary_identifier": "geonlp:geoshape-city",
          "entry_id": "13229A2001",
          "geolod_id": "pueGMO",
          "hypernym": [
            "東京都"
          ],
          "latitude": "35.72549900",
          "longitude": "139.53815900",
          "ne_class": "市区町村",
          "prefname": "東京都",
          "prefname_variants": "東京都",
          "source": "1/西東京市役所/西東京市南町 5-6-13/P34-14_13.xml",
          "suffix": [
            "市"
          ],
          "valid_from": "2001-01-21",
          "valid_to": ""
        },
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "西東京市",
          "pos": "名詞",
          "prononciation": "",
```


(次のページに続く)

(前のページからの続き)


```
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "pueGMO:西東京市",
        "surface": "西東京市",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "西東京市"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "に",
        "pos": "助詞",
        "prononciation": "ニ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "に",
        "yomi": "ニ"
      },
      "node_type": "NORMAL",
      "surface": "に"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "五段・ラ行",
        "conjugation_type": "連用形",
        "original_form": "なる",
        "pos": "動詞",
```


(次のページに続く)

(前のページからの続き)


```
          "prononciation": "ナリ",
          "subclass1": "自立",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "なり",
          "yomi": "ナリ"
        },
        "node_type": "NORMAL",
        "surface": "なり"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "特殊・マス",
          "conjugation_type": "連用形",
          "original_form": "ます",
          "pos": "助動詞",
          "prononciation": "マシ",
          "subclass1": "*",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "まし",
          "yomi": "マシ"
        },
        "node_type": "NORMAL",
        "surface": "まし"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "特殊・タ",
          "conjugation_type": "基本形",
          "original_form": "た",
```


(次のページに続く)

(前のページからの続き)


```
          "pos": "助動詞",
          "prononciation": "タ",
          "subclass1": "*",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "た",
          "yomi": "タ"
        },
        "node_type": "NORMAL",
        "surface": "た"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```

7.6.11 time-before オプション

`TimeBeforeFilter` を作成し、 *geonlp.parse* および *geonlp.parseStructured* の結果に対してこのフィルタを適用します。

パラメータで指定された時点または期間の開始日時より前に存在した地名語以外は地名語ではない固有名詞に置き換えられます。

期間の終了日時は結果に影響を与えません。

パラメータ

time-before : str, list of str, dict

- str の場合、一時点を表す日付または日時と解釈します。
- list の場合、要素が 1 つならば一時点を、2 つならば期間を表す日付または日時と解釈します。
- dict の場合、 `date_from` の値が期間の開始時点を、`date_to` の値が期間の終了時点を表す日付または日時と解釈します。 `date_to` は省略可能です。

リクエストの例

2000 年 1 月 1 日以前に存在した地名語だけを抽出します。「田無市」「保谷市」はこの時点より後に合併して「西東京市」になったので、「田無市」「保谷市」は地名語、「西東京市」は非地名語として解釈されます。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "田無市と保谷市は 2001 年 1 月 21 日に合併して西東京市になりました。",
    "options": {
      "time-before": [
        "2000-01-01",
        "2001-02-01"
      ]
    }
  },
  "id": "test_time_before"
}
```

レスポンスの例


```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          139.538159,
          35.725499
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
          "address": "東京都田無市",
          "body": "田無",
          "body_variants": "田無",
          "code": {},
          "countyname": "",
          "countyname_variants": "",
          "dictionary_id": 1,
          "dictionary_identifier": "geonlp:geoshape-city",
          "entry_id": "13216A1968",
          "geolod_id": "NGQsEB",
          "hypernym": [
            "東京都"
          ],
          "latitude": "35.72549900",
          "longitude": "139.53815900",
          "ne_class": "市区町村",
          "prefname": "東京都",
          "prefname_variants": "東京都",
          "source": "1/西東京市役所/西東京市南町 5-6-13/P34-14_13.xml/(20001001\\13\\
↪13216A1968.wkt.txt)",
          "suffix": [
            "市"
          ],
          "valid_from": "",
          "valid_to": "2001-01-21"
        },
```


(次のページに続く)

(前のページからの続き)


```
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "田無市",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "地名語",
          "subclass3": "NGQsEB:田無市",
          "surface": "田無市",
          "yomi": ""
        },
        "node_type": "GEOWORD",
        "surface": "田無市"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "と",
          "pos": "助詞",
          "prononciation": "ト",
          "subclass1": "並立助詞",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "と",
          "yomi": "ト"
        },
        "node_type": "NORMAL",
        "surface": "と"
      },
      "type": "Feature"
    },
    {
      "geometry": {
```


(次のページに続く)

(前のページからの続き)


```
        "coordinates": [
          139.558901,
          35.741527
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
          "address": "東京都保谷市",
          "body": "保谷",
          "body_variants": "保谷",
          "code": {},
          "countyname": "",
          "countyname_variants": "",
          "dictionary_id": 1,
          "dictionary_identifier": "geonlp:geoshape-city",
          "entry_id": "13217A1968",
          "geolod_id": "PuWcAj",
          "hypernym": [
            "東京都"
          ],
          "latitude": "35.74152700",
          "longitude": "139.55890100",
          "ne_class": "市区町村",
          "prefname": "東京都",
          "prefname_variants": "東京都",
          "source": "2/保谷庁舎/西東京市中町 1-5-1/P34-14_13.xml/(20001001\\13\\13217A1968.
↪wkt.txt)",
          "suffix": [
            "市"
          ],
          "valid_from": "",
          "valid_to": "2001-01-21"
        },
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "保谷市",
          "pos": "名詞",
```


(次のページに続く)

(前のページからの続き)


```
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "PuWcAj:保谷市",
        "surface": "保谷市",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "保谷市"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "は",
        "pos": "助詞",
        "prononciation": "ワ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "は",
        "yomi": "ハ"
      },
      "node_type": "NORMAL",
      "surface": "は"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "2001",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "2001"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "年",
        "pos": "名詞",
        "prononciation": "ネン",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "年",
        "yomi": "ネン"
      },
      "node_type": "NORMAL",
      "surface": "年"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "1",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "1"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "月",
        "pos": "名詞",
        "prononciation": "ツキ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "月",
        "yomi": "ツキ"
      },
      "node_type": "NORMAL",
      "surface": "月"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "21",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "21"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "日",
        "pos": "名詞",
        "prononciation": "ニチ",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "日",
        "yomi": "ニチ"
      },
      "node_type": "NORMAL",
      "surface": "日"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
```


(次のページに続く)

(前のページからの続き)


```
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "に",
        "pos": "助詞",
        "prononciation": "ニ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "に",
        "yomi": "ニ"
      },
      "node_type": "NORMAL",
      "surface": "に"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "合併",
        "pos": "名詞",
        "prononciation": "ガッペイ",
        "subclass1": "サ変接続",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "合併",
        "yomi": "ガッペイ"
      },
      "node_type": "NORMAL",
      "surface": "合併"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
```


(次のページに続く)

(前のページからの続き)


```
      "morphemes": {
        "conjugated_form": "サ変・スル",
        "conjugation_type": "連用形",
        "original_form": "する",
        "pos": "動詞",
        "prononciation": "シ",
        "subclass1": "自立",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "し",
        "yomi": "シ"
      },
      "node_type": "NORMAL",
      "surface": "し"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "て",
        "pos": "助詞",
        "prononciation": "テ",
        "subclass1": "接続助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "て",
        "yomi": "テ"
      },
      "node_type": "NORMAL",
      "surface": "て"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
```


(次のページに続く)

(前のページからの続き)


```
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "西東京市",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地域",
        "subclass3": "一般",
        "surface": "西東京市",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "西東京市"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "に",
        "pos": "助詞",
        "prononciation": "ニ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "に",
        "yomi": "ニ"
      },
      "node_type": "NORMAL",
      "surface": "に"
    },
    "type": "Feature"
  },
  {
```


(次のページに続く)

(前のページからの続き)


```
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "五段・ラ行",
        "conjugation_type": "連用形",
        "original_form": "なる",
        "pos": "動詞",
        "prononciation": "ナリ",
        "subclass1": "自立",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "なり",
        "yomi": "ナリ"
      },
      "node_type": "NORMAL",
      "surface": "なり"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・マス",
        "conjugation_type": "連用形",
        "original_form": "ます",
        "pos": "助動詞",
        "prononciation": "マシ",
        "subclass1": "*",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "まし",
        "yomi": "マシ"
      },
      "node_type": "NORMAL",
      "surface": "まし"
    },
    "type": "Feature"
  },
```


(次のページに続く)

(前のページからの続き)


```
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・タ",
        "conjugation_type": "基本形",
        "original_form": "た",
        "pos": "助動詞",
        "prononciation": "タ",
        "subclass1": "*",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "た",
        "yomi": "タ"
      },
      "node_type": "NORMAL",
      "surface": "た"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "。",
        "pos": "記号",
        "prononciation": "。",
        "subclass1": "句点",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "。",
        "yomi": "。"
      },
      "node_type": "NORMAL",
      "surface": "。"
    },
    "type": "Feature"
```


(次のページに続く)

(前のページからの続き)

```
    }
  ],
  "type": "FeatureCollection"
}
```

7.6.12 time-after オプション

`TimeAfterFilter` を作成し、 *geonlp.parse* および *geonlp.parseStructured* の結果に対してこのフィルタを適用します。

パラメータで指定された時点または期間の終了日時より後に存在した地名語以外は地名語ではない固有名詞に置き換えられます。

期間の開始日時は結果に影響を与えません。

パラメータ

time-after : str, list of str, dict

- str の場合、一時点を表す日付または日時と解釈します。
- list の場合、要素が 1 つならば一時点を、2 つならば期間を表す日付または日時と解釈します。
- dict の場合、 `date_from` の値が期間の開始時点を、`date_to` の値が期間の終了時点を表す日付または日時と解釈します。 `date_to` は省略可能です。

リクエストの例

2001 年 2 月 1 日より後に存在した地名語だけを抽出します。「田無市」「保谷市」はこの時点より前に合併して「西東京市」になったので、「田無市」「保谷市」は非地名語、「西東京市」は地名語として解釈されます。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "田無市と保谷市は 2001 年 1 月 21 日に合併して西東京市になりました。",
    "options": {
      "time-after": [
        "2000-01-01",
        "2001-02-01"
      ]
    }
  },
```

(次のページに続く)

(前のページからの続き)

```
  "id": "test_time_after"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "田無市",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "地域",
          "subclass3": "一般",
          "surface": "田無市",
          "yomi": ""
        },
        "node_type": "NORMAL",
        "surface": "田無市"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "と",
          "pos": "助詞",
          "prononciation": "ト",
          "subclass1": "並立助詞",
          "subclass2": "*",
```

(次のページに続く)

(前のページからの続き)


```
        "subclass3": "*",
        "surface": "と",
        "yomi": "ト"
      },
      "node_type": "NORMAL",
      "surface": "と"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "保谷市",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地域",
        "subclass3": "一般",
        "surface": "保谷市",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "保谷市"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "は",
        "pos": "助詞",
        "prononciation": "ワ",
        "subclass1": "係助詞",
```


(次のページに続く)

(前のページからの続き)


```
          "subclass2": "*",
          "subclass3": "*",
          "surface": "は",
          "yomi": "ハ"
        },
        "node_type": "NORMAL",
        "surface": "は"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "*",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "数",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "2001",
          "yomi": ""
        },
        "node_type": "NORMAL",
        "surface": "2001"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "年",
          "pos": "名詞",
          "prononciation": "ネン",
```


(次のページに続く)

(前のページからの続き)


```
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "年",
        "yomi": "ネン"
      },
      "node_type": "NORMAL",
      "surface": "年"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "1",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "1"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "月",
        "pos": "名詞",
```


(次のページに続く)

(前のページからの続き)


```
        "prononciation": "ツキ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "月",
        "yomi": "ツキ"
      },
      "node_type": "NORMAL",
      "surface": "月"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "21",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "21"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "日",
```


(次のページに続く)

(前のページからの続き)


```
        "pos": "名詞",
        "prononciation": "ニチ",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "日",
        "yomi": "ニチ"
      },
      "node_type": "NORMAL",
      "surface": "日"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "に",
        "pos": "助詞",
        "prononciation": "ニ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "に",
        "yomi": "ニ"
      },
      "node_type": "NORMAL",
      "surface": "に"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "original_form": "合併",
        "pos": "名詞",
        "prononciation": "ガッペイ",
        "subclass1": "サ変接続",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "合併",
        "yomi": "ガッペイ"
      },
      "node_type": "NORMAL",
      "surface": "合併"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "サ変・スル",
        "conjugation_type": "連用形",
        "original_form": "する",
        "pos": "動詞",
        "prononciation": "シ",
        "subclass1": "自立",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "し",
        "yomi": "シ"
      },
      "node_type": "NORMAL",
      "surface": "し"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "conjugation_type": "*",
        "original_form": "て",
        "pos": "助詞",
        "prononciation": "テ",
        "subclass1": "接続助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "て",
        "yomi": "テ"
      },
      "node_type": "NORMAL",
      "surface": "て"
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [
        139.538159,
        35.725499
      ],
      "type": "Point"
    },
    "properties": {
      "geoword_properties": {
        "address": "東京都西東京市",
        "body": "西東京",
        "body_variants": "西東京",
        "code": {},
        "countyname": "",
        "countyname_variants": "",
        "dictionary_id": 1,
        "dictionary_identifier": "geonlp:geoshape-city",
        "entry_id": "13229A2001",
        "geolod_id": "pueGMO",
        "hypernym": [
          "東京都"
        ],
        "latitude": "35.72549900",
```


(次のページに続く)

(前のページからの続き)


```
        "longitude": "139.53815900",
        "ne_class": "市区町村",
        "prefname": "東京都",
        "prefname_variants": "東京都",
        "source": "1/西東京市役所/西東京市南町 5-6-13/P34-14_13.xml",
        "suffix": [
          "市"
        ],
        "valid_from": "2001-01-21",
        "valid_to": ""
      },
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "西東京市",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "pueGMO:西東京市",
        "surface": "西東京市",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "西東京市"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "に",
        "pos": "助詞",
        "prononciation": "ニ",
        "subclass1": "格助詞",
        "subclass2": "一般",
```


(次のページに続く)

(前のページからの続き)


```
          "subclass3": "*",
          "surface": "に",
          "yomi": "ニ"
        },
        "node_type": "NORMAL",
        "surface": "に"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "五段・ラ行",
          "conjugation_type": "連用形",
          "original_form": "なる",
          "pos": "動詞",
          "prononciation": "ナリ",
          "subclass1": "自立",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "なり",
          "yomi": "ナリ"
        },
        "node_type": "NORMAL",
        "surface": "なり"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "特殊・マス",
          "conjugation_type": "連用形",
          "original_form": "ます",
          "pos": "助動詞",
          "prononciation": "マシ",
          "subclass1": "*",
```


(次のページに続く)

(前のページからの続き)


```
          "subclass2": "*",
          "subclass3": "*",
          "surface": "まし",
          "yomi": "マシ"
        },
        "node_type": "NORMAL",
        "surface": "まし"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "特殊・タ",
          "conjugation_type": "基本形",
          "original_form": "た",
          "pos": "助動詞",
          "prononciation": "タ",
          "subclass1": "*",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "た",
          "yomi": "タ"
        },
        "node_type": "NORMAL",
        "surface": "た"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
```


(次のページに続く)

(前のページからの続き)

```
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```

7.6.13 time-overlaps オプション

`TimeOverlapsFilter` を作成し、 *geonlp.parse* および *geonlp.parseStructured* の結果に対してこのフィルタを適用します。

パラメータで指定された時点または期間に存在した地名語以外は地名語ではない固有名詞に置き換えられます。

このオプションは *time-exists* **オプション**と同じです。

パラメータ

time-overlaps : str, list of str, dict

- str の場合、一時点を表す日付または日時と解釈します。
- list の場合、要素が 1 つならば一時点を、2 つならば期間を表す日付または日時と解釈します。
- dict の場合、 `date_from` の値が期間の開始時点を、`date_to` の値が期間の終了時点を表す日付または日時と解釈します。 `date_to` は省略可能です。

リクエストの例

2000 年 1 月 1 日から 2001 年 2 月 1 日の間に存在した地名語だけを抽出します。「田無市」「保谷市」はこの期間内に合併して「西東京市」になったので、「田無市」「保谷市」「西東京市」は地名語として解釈されます。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "田無市と保谷市は 2001 年 1 月 21 日に合併して西東京市になりました。",
    "options": {
      "time-overlaps": [
        "2000-01-01",
        "2001-02-01"
      ]
    }
  },
  "id": "test_time_overlaps"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          139.538159,
          35.725499
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
          "address": "東京都田無市",
          "body": "田無",
          "body_variants": "田無",
          "code": {},
          "countyname": "",
          "countyname_variants": "",
          "dictionary_id": 1,
```

(次のページに続く)

(前のページからの続き)


```
          "dictionary_identifier": "geonlp:geoshape-city",
          "entry_id": "13216A1968",
          "geolod_id": "NGQsEB",
          "hypernym": [
            "東京都"
          ],
          "latitude": "35.72549900",
          "longitude": "139.53815900",
          "ne_class": "市区町村",
          "prefname": "東京都",
          "prefname_variants": "東京都",
          "source": "1/西東京市役所/西東京市南町 5-6-13/P34-14_13.xml/(20001001\\13\\
↪13216A1968.wkt.txt)",
          "suffix": [
            "市"
          ],
          "valid_from": "",
          "valid_to": "2001-01-21"
        },
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "田無市",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "地名語",
          "subclass3": "NGQsEB:田無市",
          "surface": "田無市",
          "yomi": ""
        },
        "node_type": "GEOWORD",
        "surface": "田無市"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
```


(次のページに続く)

(前のページからの続き)


```
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "と",
          "pos": "助詞",
          "prononciation": "ト",
          "subclass1": "並立助詞",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "と",
          "yomi": "ト"
        },
        "node_type": "NORMAL",
        "surface": "と"
      },
      "type": "Feature"
    },
    {
      "geometry": {
        "coordinates": [
          139.558901,
          35.741527
        ],
        "type": "Point"
      },
      "properties": {
        "geoword_properties": {
          "address": "東京都保谷市",
          "body": "保谷",
          "body_variants": "保谷",
          "code": {},
          "countyname": "",
          "countyname_variants": "",
          "dictionary_id": 1,
          "dictionary_identifier": "geonlp:geoshape-city",
          "entry_id": "13217A1968",
          "geolod_id": "PuWcAj",
          "hypernym": [
            "東京都"
```


(次のページに続く)

(前のページからの続き)


```
          ],
          "latitude": "35.74152700",
          "longitude": "139.55890100",
          "ne_class": "市区町村",
          "prefname": "東京都",
          "prefname_variants": "東京都",
          "source": "2/保谷庁舎/西東京市中町 1-5-1/P34-14_13.xml/(20001001\\13\\13217A1968.
↪wkt.txt)",
          "suffix": [
            "市"
          ],
          "valid_from": "",
          "valid_to": "2001-01-21"
        },
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "保谷市",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "地名語",
          "subclass3": "PuWcAj:保谷市",
          "surface": "保谷市",
          "yomi": ""
        },
        "node_type": "GEOWORD",
        "surface": "保谷市"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "は",
          "pos": "助詞",
```


(次のページに続く)

(前のページからの続き)


```
        "prononciation": "ワ",
        "subclass1": "係助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "は",
        "yomi": "ハ"
      },
      "node_type": "NORMAL",
      "surface": "は"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "2001",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "2001"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "年",
```


(次のページに続く)

(前のページからの続き)


```
        "pos": "名詞",
        "prononciation": "ネン",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "年",
        "yomi": "ネン"
      },
      "node_type": "NORMAL",
      "surface": "年"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "1",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "1"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "original_form": "月",
        "pos": "名詞",
        "prononciation": "ツキ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "月",
        "yomi": "ツキ"
      },
      "node_type": "NORMAL",
      "surface": "月"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "21",
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "21"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "conjugation_type": "*",
        "original_form": "日",
        "pos": "名詞",
        "prononciation": "ニチ",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "日",
        "yomi": "ニチ"
      },
      "node_type": "NORMAL",
      "surface": "日"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "に",
        "pos": "助詞",
        "prononciation": "ニ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "に",
        "yomi": "ニ"
      },
      "node_type": "NORMAL",
      "surface": "に"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
```


(次のページに続く)

(前のページからの続き)


```
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "合併",
        "pos": "名詞",
        "prononciation": "ガッペイ",
        "subclass1": "サ変接続",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "合併",
        "yomi": "ガッペイ"
      },
      "node_type": "NORMAL",
      "surface": "合併"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "サ変・スル",
        "conjugation_type": "連用形",
        "original_form": "する",
        "pos": "動詞",
        "prononciation": "シ",
        "subclass1": "自立",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "し",
        "yomi": "シ"
      },
      "node_type": "NORMAL",
      "surface": "し"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
```


(次のページに続く)

(前のページからの続き)


```
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "て",
        "pos": "助詞",
        "prononciation": "テ",
        "subclass1": "接続助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "て",
        "yomi": "テ"
      },
      "node_type": "NORMAL",
      "surface": "て"
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [
        139.538159,
        35.725499
      ],
      "type": "Point"
    },
    "properties": {
      "geoword_properties": {
        "address": "東京都西東京市",
        "body": "西東京",
        "body_variants": "西東京",
        "code": {},
        "countyname": "",
        "countyname_variants": "",
        "dictionary_id": 1,
        "dictionary_identifier": "geonlp:geoshape-city",
        "entry_id": "13229A2001",
        "geolod_id": "pueGMO",
        "hypernym": [
          "東京都"
```


(次のページに続く)

(前のページからの続き)


```
        ],
        "latitude": "35.72549900",
        "longitude": "139.53815900",
        "ne_class": "市区町村",
        "prefname": "東京都",
        "prefname_variants": "東京都",
        "source": "1/西東京市役所/西東京市南町 5-6-13/P34-14_13.xml",
        "suffix": [
          "市"
        ],
        "valid_from": "2001-01-21",
        "valid_to": ""
      },
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "西東京市",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地名語",
        "subclass3": "pueGMO:西東京市",
        "surface": "西東京市",
        "yomi": ""
      },
      "node_type": "GEOWORD",
      "surface": "西東京市"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "に",
        "pos": "助詞",
        "prononciation": "ニ",
```


(次のページに続く)

(前のページからの続き)


```
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
        "surface": "に",
        "yomi": "ニ"
      },
      "node_type": "NORMAL",
      "surface": "に"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "五段・ラ行",
        "conjugation_type": "連用形",
        "original_form": "なる",
        "pos": "動詞",
        "prononciation": "ナリ",
        "subclass1": "自立",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "なり",
        "yomi": "ナリ"
      },
      "node_type": "NORMAL",
      "surface": "なり"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・マス",
        "conjugation_type": "連用形",
        "original_form": "ます",
        "pos": "助動詞",
```


(次のページに続く)

(前のページからの続き)


```
        "prononciation": "マシ",
        "subclass1": "*",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "まし",
        "yomi": "マシ"
      },
      "node_type": "NORMAL",
      "surface": "まし"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・タ",
        "conjugation_type": "基本形",
        "original_form": "た",
        "pos": "助動詞",
        "prononciation": "タ",
        "subclass1": "*",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "た",
        "yomi": "タ"
      },
      "node_type": "NORMAL",
      "surface": "た"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "。",
```


(次のページに続く)

(前のページからの続き)

```
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```

7.6.14 time-covers オプション

`TimeCoversFilter` を作成し、 *geonlp.parse* および *geonlp.parseStructured* の結果に対してこのフィルタを適用します。

パラメータで指定された時点または期間の開始日時から終了日時まで存在し続けた地名語以外は地名語ではない固有名詞に置き換えられます。

パラメータ

time-covers : str, list of str, dict

- str の場合、一時点を表す日付または日時と解釈します。
- list の場合、要素が 1 つならば一時点を、2 つならば期間を表す日付または日時と解釈します。
- dict の場合、 `date_from` の値が期間の開始時点を、`date_to` の値が期間の終了時点を表す日付または日時と解釈します。 `date_to` は省略可能です。

リクエストの例

2000 年 1 月 1 日から 2001 年 2 月 1 日の間、ずっと存在した地名語だけを抽出します。「田無市」「保谷市」はこの期間内に合併して「西東京市」になったので、「田無市」「保谷市」「西東京市」は非地名語として解釈されます。

```
{
  "method": "geonlp.parse",
  "params": {
    "sentence": "田無市と保谷市は 2001 年 1 月 21 日に合併して西東京市になりました。",
    "options": {
      "time-covers": [
        "2000-01-01",
        "2001-02-01"
      ]
    }
  },
  "id": "test_time_covers"
}
```

レスポンスの例

```
{
  "features": [
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "田無市",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "地域",
          "subclass3": "一般",
          "surface": "田無市",
          "yomi": ""
        },
        "node_type": "NORMAL",
        "surface": "田無市"
```

(次のページに続く)

(前のページからの続き)


```
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "と",
        "pos": "助詞",
        "prononciation": "ト",
        "subclass1": "並立助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "と",
        "yomi": "ト"
      },
      "node_type": "NORMAL",
      "surface": "と"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "保谷市",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
        "subclass2": "地域",
        "subclass3": "一般",
        "surface": "保谷市",
        "yomi": ""
      },
      "node_type": "NORMAL",
```


(次のページに続く)

(前のページからの続き)


```
        "surface": "保谷市"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "は",
          "pos": "助詞",
          "prononciation": "ワ",
          "subclass1": "係助詞",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "は",
          "yomi": "ハ"
        },
        "node_type": "NORMAL",
        "surface": "は"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "*",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "数",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "2001",
          "yomi": ""
        },
```


(次のページに続く)

(前のページからの続き)


```
      "node_type": "NORMAL",
      "surface": "2001"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "年",
        "pos": "名詞",
        "prononciation": "ネン",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "年",
        "yomi": "ネン"
      },
      "node_type": "NORMAL",
      "surface": "年"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "1",
        "yomi": ""
```


(次のページに続く)

(前のページからの続き)


```
      },
      "node_type": "NORMAL",
      "surface": "1"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "月",
        "pos": "名詞",
        "prononciation": "ツキ",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "月",
        "yomi": "ツキ"
      },
      "node_type": "NORMAL",
      "surface": "月"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "21",
```


(次のページに続く)

(前のページからの続き)


```
        "yomi": ""
      },
      "node_type": "NORMAL",
      "surface": "21"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "日",
        "pos": "名詞",
        "prononciation": "ニチ",
        "subclass1": "接尾",
        "subclass2": "助数詞",
        "subclass3": "*",
        "surface": "日",
        "yomi": "ニチ"
      },
      "node_type": "NORMAL",
      "surface": "日"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "に",
        "pos": "助詞",
        "prononciation": "ニ",
        "subclass1": "格助詞",
        "subclass2": "一般",
        "subclass3": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "surface": "に",
        "yomi": "ニ"
      },
      "node_type": "NORMAL",
      "surface": "に"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "合併",
        "pos": "名詞",
        "prononciation": "ガッペイ",
        "subclass1": "サ変接続",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "合併",
        "yomi": "ガッペイ"
      },
      "node_type": "NORMAL",
      "surface": "合併"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "サ変・スル",
        "conjugation_type": "連用形",
        "original_form": "する",
        "pos": "動詞",
        "prononciation": "シ",
        "subclass1": "自立",
        "subclass2": "*",
```


(次のページに続く)

(前のページからの続き)


```
        "subclass3": "*",
        "surface": "し",
        "yomi": "シ"
      },
      "node_type": "NORMAL",
      "surface": "し"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "て",
        "pos": "助詞",
        "prononciation": "テ",
        "subclass1": "接続助詞",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "て",
        "yomi": "テ"
      },
      "node_type": "NORMAL",
      "surface": "て"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "西東京市",
        "pos": "名詞",
        "prononciation": "",
        "subclass1": "固有名詞",
```


(次のページに続く)

(前のページからの続き)


```
          "subclass2": "地域",
          "subclass3": "一般",
          "surface": "西東京市",
          "yomi": ""
        },
        "node_type": "NORMAL",
        "surface": "西東京市"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "に",
          "pos": "助詞",
          "prononciation": "ニ",
          "subclass1": "格助詞",
          "subclass2": "一般",
          "subclass3": "*",
          "surface": "に",
          "yomi": "ニ"
        },
        "node_type": "NORMAL",
        "surface": "に"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "五段・ラ行",
          "conjugation_type": "連用形",
          "original_form": "なる",
          "pos": "動詞",
          "prononciation": "ナリ",
```


(次のページに続く)

(前のページからの続き)


```
        "subclass1": "自立",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "なり",
        "yomi": "ナリ"
      },
      "node_type": "NORMAL",
      "surface": "なり"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・マス",
        "conjugation_type": "連用形",
        "original_form": "ます",
        "pos": "助動詞",
        "prononciation": "マシ",
        "subclass1": "*",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "まし",
        "yomi": "マシ"
      },
      "node_type": "NORMAL",
      "surface": "まし"
    },
    "type": "Feature"
  },
  {
    "geometry": null,
    "properties": {
      "morphemes": {
        "conjugated_form": "特殊・タ",
        "conjugation_type": "基本形",
        "original_form": "た",
        "pos": "助動詞",
```


(次のページに続く)

(前のページからの続き)


```
          "prononciation": "タ",
          "subclass1": "*",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "た",
          "yomi": "タ"
        },
        "node_type": "NORMAL",
        "surface": "た"
      },
      "type": "Feature"
    },
    {
      "geometry": null,
      "properties": {
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "。",
          "pos": "記号",
          "prononciation": "。",
          "subclass1": "句点",
          "subclass2": "*",
          "subclass3": "*",
          "surface": "。",
          "yomi": "。"
        },
        "node_type": "NORMAL",
        "surface": "。"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```

第 8 章

解析方法のチューニング

解析結果に含まれる地名語が想定と異なる場合、解析方法をチューニングする必要があります。

想定した地名語が得られないのは、そもそもその地名語がデータベースに登録されていない場合と、表記が同じ別の地名語が採用されている（＝地名解決処理が失敗している）場合の二通りがあります。

想定した地名語がデータベースに登録されていない場合は、その地名語が含まれている**拡張辞書をインストール**してください。

ここでは地名解決処理が失敗してしまう場合に、処理精度を改善する方法を説明します。

8.1 特定の語を抽出しない

環境変数 **GEONLP_EXCLUDED_WORD** に指定した語は地名語として解析されなくなり、MeCab の解析結果がそのまま出力されます。

```
$ export GEONLP_EXCLUDED_WORD=甲子園
$ echo "甲子園に行こう。" | pygeonlp geoparse
甲子園　名詞,固有名詞,地域,一般,*,*,甲子園,コウシエン,コーシエン
に　　助詞,格助詞,一般,*,*,*,に,ニ,ニ
行こ　　動詞,自立,*,*,未然ウ接続,五段・カ行促音便,行く,イコ,イコ
う　　助動詞,*,*,*,基本形,不変化型,う,ウ,ウ
。　　記号,句点,*,*,*,*,。,。,。
EOS
```

複数の語を指定したい場合は | を区切りとして列挙してください。

```
$ export GEONLP_EXCLUDED_WORD='神戸|広島'
$ echo "今年の J1 は神戸と広島が強い。" | pygeonlp geoparse
今年　　名詞,副詞可能,*,*,*,*,今年,コトシ,コトシ
```

(次のページに続く)

(前のページからの続き)

```
の      助詞,連体化,*,*,*,*,の,ノ,ノ
J       名詞,一般,*,*,*,*,*,,
1       名詞,数,*,*,*,*,*,,
は      助詞,係助詞,*,*,*,*,は,ハ,ワ
神戸    名詞,固有名詞,地域,一般,*,*,神戸,コウベ,コーベ
と      助詞,並立助詞,*,*,*,*,と,ト,ト
広島    名詞,固有名詞,組織,*,*,*,広島,ヒロシマ,ヒロシマ
が      助詞,格助詞,一般,*,*,*,が,ガ,ガ
強い    形容詞,自立,*,*,基本形,形容詞・アウオ段,強い,ツヨイ,ツヨイ
。      記号,句点,*,*,*,*,。,。,。
EOS
```

8.2 解析対象とする地名解析辞書・固有名クラスの指定

市区町村名だけを抽出したいのに駅名が抽出されてしまうといった場合、*disactivateDictionaries()* を実行することで、一時的に特定の地名解析辞書を利用しないように設定することができます。

例：「和歌山市」のつもりが「和歌山市駅」が抽出されてしまう。

```
>>> import pygeonlp.api as api
>>> api.geoparse('和歌山市は晴れ')
[{'type': 'Feature', 'geometry': {'type': 'Point', 'coordinates': [135.16538500000001,␣
↪34.23604]}, 'properties': {'surface': '和歌山市', 'node_type': 'GEOWORD', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '和歌山市', 'pos': '名
詞', 'prononciation': '', 'subclass1': '固有名詞', 'subclass2': '地名語', 'subclass3':
↪'OciY0C:和歌山市駅', 'surface': '和歌山市', 'yomi': ''}, 'geoword_properties': {'body': '
和歌山市', 'dictionary_id': 3, 'entry_id': 'adeb575da6e2879b67c9b76d269333e6', 'geolod_id
↪': 'OciY0C', 'hypernym': ['南海電気鉄道', '和歌山港線'], 'institution_type': '民営鉄道',
 'latitude': '34.23604', 'longitude': '135.16538500000001', 'ne_class': '鉄道施設/鉄道駅',
↪'railway_class': '普通鉄道', 'suffix': ['駅', ''], 'dictionary_identifier': 'geonlp:ksj-
↪station-N02-2019'}}}, ... ]
```

次のコードでは、辞書の identifier に **station** を含むものを一時的に利用しないように指定します。その結果 **geonlp:ksj-station-N02-2019** が利用できなくなるので、和歌山市駅が候補から除外されます。

```
>>> import pygeonlp.api as api
>>> api.disactivateDictionaries(pattern=r'station')
>>> api.geoparse('和歌山市は晴れ。')
[{'type': 'Feature', 'geometry': {'type': 'Point', 'coordinates': [135.170808, 34.
```

(次のページに続く)

(前のページからの続き)

```
↪230514]}, 'properties': {'surface': '和歌山市', 'node_type': 'GEOWORD', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '和歌山市', 'pos': '名
詞', 'prononciation': '', 'subclass1': '固有名詞', 'subclass2': '地名語', 'subclass3':
↪'lQccqK:和歌山市', 'surface': '和歌山市', 'yomi': ''}, 'geoword_properties': {'address
↪': '和歌山県和歌山市', 'body': '和歌山', 'body_variants': '和歌山', 'code': {},
 'countyname': '', 'countyname_variants': '', 'dictionary_id': 1, 'entry_id': '30201A1968
↪', 'geolod_id': 'lQccqK', 'hypernym': ['和歌山県'], 'latitude': '34.23051400',
↪'longitude': '135.17080800', 'ne_class': '市区町村', 'prefname': '和歌山県', 'prefname_
↪variants': '和歌山県', 'source': '1/和歌山市役所/和歌山市七番丁 23/P34-14_30.xml', 'suffix
↪': ['市'], 'valid_from': '1889-04-01', 'valid_to': '', 'dictionary_identifier':
↪'geonlp:geoshape-city'}}}, ... ]
```

辞書ごと対象から除外する代わりに、:py:meth:~pygeonlp.api.setActiveClasses` で対象とする固有名クラスを指定することで、クラス単位で抽出対象を限定することもできます。

次のコードでは全ての固有名クラス (**r'.*'**) から鉄道施設を除外 (**r'-鉄道施設/.*'**) したクラスを抽出対象にすることで、和歌山市駅を候補から除外します。

```
>>> import pygeonlp.api as api
>>> api.setActiveClasses(patterns=[r'.*', r'-鉄道施設/.*'])
>>> api.geoparse('和歌山市は晴れ。')
[{'type': 'Feature', 'geometry': {'type': 'Point', 'coordinates': [135.170808, 34.
↪230514]}, 'properties': {'surface': '和歌山市', 'node_type': 'GEOWORD', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '和歌山市', 'pos': '名
詞', 'prononciation': '', 'subclass1': '固有名詞', 'subclass2': '地名語', 'subclass3':
↪'lQccqK:和歌山市', 'surface': '和歌山市', 'yomi': ''}, 'geoword_properties': {'address
↪': '和歌山県和歌山市', 'body': '和歌山', 'body_variants': '和歌山', 'code': {},
 'countyname': '', 'countyname_variants': '', 'dictionary_id': 1, 'entry_id': '30201A1968
↪', 'geolod_id': 'lQccqK', 'hypernym': ['和歌山県'], 'latitude': '34.23051400',
↪'longitude': '135.17080800', 'ne_class': '市区町村', 'prefname': '和歌山県', 'prefname_
↪variants': '和歌山県', 'source': '1/和歌山市役所/和歌山市七番丁 23/P34-14_30.xml', 'suffix
↪': ['市'], 'valid_from': '1889-04-01', 'valid_to': '', 'dictionary_identifier':
↪'geonlp:geoshape-city'}}}, ... ]
```

8.3 地名語抽出ルールの変更

地名語抽出ルールは :py:meth:pygeonlp.api.init` 実行時のオプションパラメータの一つで、形態素解析レベルの処理ルールを細かく設定します。一度設定したルールは再び init() を呼ぶまで変更できません。地名語抽出ルールについては init() のメモの欄を参照してください。

地名語抽出ルールはほとんどの場合は変更する必要がありませんが、**excluded_word** オプションは単純で、場合によっては効果的です。このオプションは、指定した地名語を地名語として抽出しないようにします。

例：「甲子園」は全国高校野球大会の意味なのに駅名として抽出されてしまう。

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.geoparse('甲子園に行こう')
[{'type': 'Feature', 'geometry': {'type': 'Point', 'coordinates': [135.363275, 34.
↪723960000000005]}, 'properties': {'surface': '甲子園', 'node_type': 'GEOWORD',
 'morphemes': {'conjugated_form': '', 'conjugation_type': '*', 'original_form': '甲子園',
↪'pos': '名詞', 'prononciation': '', 'subclass1': '固有名詞', 'subclass2': '地名語',
 'subclass3': 'M4C8N9:甲子園駅', 'surface': '甲子園', 'yomi': ''}, 'geoword_properties': {
↪'body': '甲子園', 'dictionary_id': 3, 'entry_id': '2670a9643e77eebd8397a3236ff90514',
 'geolod_id': 'M4C8N9', 'hypernym': ['阪神電気鉄道', '本線'], 'institution_type': '民営鉄道
', 'latitude': '34.723960000000005', 'longitude': '135.363275', 'ne_class': '鉄道施設/鉄道
駅', 'railway_class': '普通鉄道', 'suffix': ['駅', ''], 'dictionary_identifier':
↪'geonlp:ksj-station-N02-2019'}}},  ... ]
```

次のコードでは **excluded_word** に「甲子園」を指定して、地名語ではなく固有名詞として抽出します。

```
>>> import pygeonlp.api as api
>>> api.init(geoword_rules={'excluded_word':['甲子園']})
>>> api.geoparse('甲子園に行こう')
[{'type': 'Feature', 'geometry': None, 'properties': {'surface': '甲子園', 'node_type':
↪'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_form
↪': '甲子園', 'pos': '名詞', 'prononciation': 'コーシエン', 'subclass1': '固有名詞',
 'subclass2': '地域', 'subclass3': '一般', 'surface': '甲子園', 'yomi': 'コウシエン'}}}, .
↪.. ]
```

8.4 フィルタの利用

対象としているテキストの時間的範囲や空間的範囲が限定されている場合（たとえば東京都内であることが分かっている場合など）は、抽出された地名語候補にフィルタ (*Filter* のサブクラス) を適用して範囲外の候補を除去することができます。

例：東京の「府中駅」のつもりが京都府の天橋立近くの「府中駅」になってしまう。

```
>>> import pygeonlp.api as api
>>> api.geoparse('府中に行きます')
[{'type': 'Feature', 'geometry': {'type': 'Point', 'coordinates': [135.195275, 35.
↪583365]}, 'properties': {'surface': '府中', 'node_type': 'GEOWORD', 'morphemes': {
↪'conjugated_form': '', 'conjugation_type': '*', 'original_form': '府中', 'pos': '名詞',
 'prononciation': '', 'subclass1': '固有名詞', 'subclass2': '地名語', 'subclass3':
↪'Auq8Kv:府中駅', 'surface': '府中', 'yomi': ''}, 'geoword_properties': {'body': '府中',
 'dictionary_id': 3, 'entry_id': 'ecabefc60f23d0442029793c6eab81d0', 'geolod_id': 'Auq8Kv
↪', 'hypernym': ['丹後海陸交通', '天橋立鋼索鉄道'], 'institution_type': '民営鉄道',
↪'latitude': '35.583365', 'longitude': '135.195275', 'ne_class': '鉄道施設/鉄道駅',
↪'railway_class': '鋼索鉄道', 'suffix': ['駅', ''], 'dictionary_identifier': 'geonlp:ksj-
↪station-N02-2019'}}}, {'type': 'Feature', 'geometry': None, 'properties': {'surface': '
に', 'node_type': 'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*',
 'original_form': 'に', 'pos': '助詞', 'prononciation': 'ニ', 'subclass1': '格助詞',
↪'subclass2': '一般', 'subclass3': '*', 'surface': 'に', 'yomi': 'ニ'}}}, ... ]
```

次のコードでは、東京都付近の四角形内に空間範囲を限定する GeoContainsFilter を適用することで、その外側にある京都府の府中駅を候補から除外し、東京都の府中駅を抽出します。

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.spatial_filter import GeoContainsFilter
>>> gcfilter = GeoContainsFilter({"type":"Polygon","coordinates":[[[139.43,35.54],[139.
↪91,35.54],[139.91,35.83],[139.43,35.83],[139.43,35.54]]]})
>>> api.geoparse('府中に行きます', filters=[gcfilter])
[{'type': 'Feature', 'geometry': {'type': 'Point', 'coordinates': [139.4801, 35.67219]},
 'properties': {'surface': '府中', 'node_type': 'GEOWORD', 'morphemes': {'conjugated_form
↪': '', 'conjugation_type': '*', 'original_form': '府中', 'pos': '名詞', 'prononciation
↪': '', 'subclass1': '固有名詞', 'subclass2': '地名語', 'subclass3': 'JQSUIi:府中駅',
 'surface': '府中', 'yomi': ''}, 'geoword_properties': {'body': '府中', 'dictionary_id':␣
↪3, 'entry_id': 'd7596c3444b3632f5236ae9e3168bab9', 'geolod_id': 'JQSUIi', 'hypernym': [
↪'京王電鉄', '京王線'], 'institution_type': '民営鉄道', 'latitude': '35.67219', 'longitude
↪': '139.4801', 'ne_class': '鉄道施設/鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅
↪', ''], 'dictionary_identifier': 'geonlp:ksj-station-N02-2019'}}}, {'type': 'Feature',
```

(次のページに続く)

(前のページからの続き)

```
↪'geometry': None, 'properties': {'surface': 'に', 'node_type': 'NORMAL', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': 'に', 'pos': '助詞',
↪'prononciation': 'ニ', 'subclass1': '格助詞', 'subclass2': '一般', 'subclass3': '*',
↪'surface': 'に', 'yomi': 'ニ'}}}, ... ]
```

フィルタはいくつでも指定できますが、複数のフィルタを指定した場合には全てのフィルタを通過する地名語だけが残ります（AND 条件）。

8.5 スコアリング方法のカスタマイズ

上述の方法は、特定の条件に合わせて個別に対応する方法です。

個別対応ではなく、組み込みの地名語選択ロジックの代わりに独自のロジックで地名語を選択したい場合には、

- パスに対するスコアを計算する関数 **path_score()**
- ノード間の関係によるスコアを計算する関数 **node_relation_score()**

を持つスコアリングクラス *pygeonlp.api.scoring.ScoringClass* からサブクラスを派生し、*geoparse()* のオプションパラメータ **scoring_class** でクラス名を指定してください。スコアリングクラスの実装については *ScoringClass* がサンプル実装となっていますので参考にしてください。

スコアを計算する関数に拡張パラメータを渡したい場合は、 **geoparse()** のオプションパラメータ **scoring_options** に任意の型の値を指定します。この値はスコアリングクラスのメンバ変数 **options** に格納されますので、ノード間のスコアを計算する関数 *node_relation_score()* およびパスのスコアを計算する関数 *path_score()* の中で **self.options** を参照して利用してください。

一例として、指定した固有名クラスの数をスコアとして返す単純なスコアリングクラスを定義し、そのスコアリングクラスを利用して geoparse の結果を表示するコードを示します。

```
"""
スコアリング方法のカスタマイズ
サンプルコード

このコードをテストするには以下のコマンドを実行してください。
python sample_myscore.py
"""
import pygeonlp.api as api
from pygeonlp.api.linker import Evaluator
from pygeonlp.api.scoring import ScoringClass
```

(次のページに続く)

(前のページからの続き)

```
api.init()


class MyScoringClass(ScoringClass):

    def path_score(self, path):
        """
        パスの中に指定した文字列で始まる固有名クラスの地名語が
        存在する数をスコアとして返すスコアリングメソッド。

        Parameters
        ----------
        path : list of Node
            解析結果候補のパス表現。
        self.options : str
            カウントする固有名クラスの先頭文字列

        Returns
        -------
        int
            target_class にマッチする固有名クラスを持つ地名語数。
        """
        if not isinstance(self.options, str):
            raise RuntimeError(
                "オプションパラメータは文字列で指定してください。")

        target_class = self.options
        score = 0
        geowords = Evaluator.collect_geowords(path)
        for geoword in geowords:
            if geoword.prop['ne_class'].startswith(target_class):
                score += 1

        return score


if __name__ == '__main__':
    print("'鉄道施設' が多い候補を優先した場合。")
    api.init(scoring_class=MyScoringClass, scoring_options='鉄道施設')
```

(次のページに続く)

(前のページからの続き)

```
    print(api.geoparse('和歌山市は晴れ。'))
    print("'市区町村' が多い候補を優先した場合。")
    api.init(scoring_class=MyScoringClass, scoring_options='市区町村')
    print(api.geoparse('和歌山市は晴れ。'))
```

実行結果は次のようになります。

```
$ python myscore.py
'鉄道施設' が多い候補を優先した場合。
[{'type': 'Feature', 'geometry': {'type': 'Point', 'coordinates': [135.165385000000001,␣
↪34.23604]}, 'properties': {'surface': '和歌山市', 'node_type': 'GEOWORD', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '和歌山市', 'pos': '名
詞', 'prononciation': '', 'subclass1': '固有名詞', 'subclass2': '地名語', 'subclass3':
↪'OciY0C:和歌山市駅', 'surface': '和歌山市', 'yomi': ''}, 'geoword_properties': {'body': '
和歌山市', 'dictionary_id': 3, 'entry_id': 'adeb575da6e2879b67c9b76d269333e6', 'geolod_id
↪': 'OciY0C', 'hypernym': ['南海電気鉄道', '和歌山港線'], 'institution_type': '民営鉄道',
 'latitude': '34.23604', 'longitude': '135.165385000000001', 'ne_class': '鉄道施設/鉄道駅',
↪'railway_class': '普通鉄道', 'suffix': ['駅', ''], 'dictionary_identifier': 'geonlp:ksj-
↪station-N02-2019'}}}, {'type': 'Feature', 'geometry': None, 'properties': {'surface': '
は', 'node_type': 'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*',
 'original_form': 'は', 'pos': '助詞', 'prononciation': 'ワ', 'subclass1': '係助詞',
↪'subclass2': '*', 'subclass3': '*', 'surface': 'は', 'yomi': 'ハ'}}}, {'type': 'Feature
↪', 'geometry': None, 'properties': {'surface': '晴れ', 'node_type': 'NORMAL',
 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '晴れ',
↪'pos': '名詞', 'prononciation': 'ハレ', 'subclass1': '一般', 'subclass2': '*',
 'subclass3': '*', 'surface': '晴れ', 'yomi': 'ハレ'}}}, {'type': 'Feature', 'geometry':␣
↪None, 'properties': {'surface': '。', 'node_type': 'NORMAL', 'morphemes': {'conjugated_
↪form': '*', 'conjugation_type': '*', 'original_form': '。', 'pos': '記号',
↪'prononciation': '。', 'subclass1': '句点', 'subclass2': '*', 'subclass3': '*', 'surface
↪': '。', 'yomi': '。'}}}]
'市区町村' が多い候補を優先した場合。
[{'type': 'Feature', 'geometry': {'type': 'Point', 'coordinates': [135.170808, 34.
↪230514]}, 'properties': {'surface': '和歌山市', 'node_type': 'GEOWORD', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '和歌山市', 'pos': '名
詞', 'prononciation': '', 'subclass1': '固有名詞', 'subclass2': '地名語', 'subclass3':
↪'lQccqK:和歌山市', 'surface': '和歌山市', 'yomi': ''}, 'geoword_properties': {'address
↪': '和歌山県和歌山市', 'body': '和歌山', 'body_variants': '和歌山', 'code': {},
 'countyname': '', 'countyname_variants': '', 'dictionary_id': 1, 'entry_id': '30201A1968
↪', 'geolod_id': 'lQccqK', 'hypernym': ['和歌山県'], 'latitude': '34.23051400',
```

(次のページに続く)

(前のページからの続き)

```
↪'longitude': '135.17080800', 'ne_class': '市区町村', 'prefname': '和歌山県', 'prefname_
↪variants': '和歌山県', 'source': '1/和歌山市役所/和歌山市七番丁23/P34-14_30.xml', 'suffix
↪': ['市'], 'valid_from': '1889-04-01', 'valid_to': '', 'dictionary_identifier':
↪'geonlp:geoshape-city'}}}, {'type': 'Feature', 'geometry': None, 'properties': {
↪'surface': 'は', 'node_type': 'NORMAL', 'morphemes': {'conjugated_form': '*',
↪'conjugation_type': '*', 'original_form': 'は', 'pos': '助詞', 'prononciation': 'ワ',
↪'subclass1': '係助詞', 'subclass2': '*', 'subclass3': '*', 'surface': 'は', 'yomi': 'ハ
'}}}, {'type': 'Feature', 'geometry': None, 'properties': {'surface': '晴れ', 'node_type
↪': 'NORMAL', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_
↪form': '晴れ', 'pos': '名詞', 'prononciation': 'ハレ', 'subclass1': '一般', 'subclass2
↪': '*', 'subclass3': '*', 'surface': '晴れ', 'yomi': 'ハレ'}}}, {'type': 'Feature',
↪'geometry': None, 'properties': {'surface': '。', 'node_type': 'NORMAL', 'morphemes': {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '。', 'pos': '記号',
↪'prononciation': '。', 'subclass1': '句点', 'subclass2': '*', 'subclass3': '*', 'surface
↪': '。', 'yomi': '。'}}}]
```

第 9 章

カスタム辞書の作成

解析したい地名語を含む地名解析辞書がウェブ上に見つからない場合は、独自の地名解析辞書を作って登録することができます。

地名解析辞書は地名語の表記とその属性のリストを列挙した CSV ファイルと、辞書の概要（メタデータ）を記述した JSON ファイルから構成されます。

9.1 カスタム辞書の例

ここではサンプルとして、「東京タワー」「東京スカイツリー」という 2 つの地名語を含む辞書を作ってみます。

まず CSV ファイルは以下のようになります。

```
entry_id,body,ne_class,latitude,longitude,url
mydic001,東京タワー,通信施設,35.65861,139.74556,https://www.tokyotower.co.jp/
mydic002,東京スカイツリー,通信施設,35.710139,139.810833,https://www.tokyo-skytree.jp/
```

CSV ファイルの必須項目は以下の 5 つです。

- entry_id: 辞書内で一意な識別子（1 文字以上の文字列）
- body: 地名語の表記（1 文字以上 255 文字以下の文字列）
- ne_class: この語がどんな種類の地名なのかを表す「固有名クラス」
- latitude: 10 進度数で表した緯度
- longitude: 10 進度数で表した経度

このサンプルでは必須項目の他、 *url* を拡張しています。

次にこの辞書の概要を示す JSON ファイルを作ります。

```
{
    "name": "カスタム地名語辞書",
    "description": "個人で利用する建物名の辞書",
    "identifier": ["geonlp:my-buildings"]
}
```

JSON ファイルの必須項目は *name* と *identifier* の 2 つです。ただし *identifier* は必ずリスト ([...]) として定義し、そのうちの 1 つは *geonlp:* から始まる必要があります。

このサンプルでは必須項目のほかに、説明用テキストを *description* に定義しています。

9.2 カスタム辞書の登録

CSV ファイルと JSON ファイルを mydic.csv, mydic.json というファイル名で保存します（ファイル名は何でも構いません）。

この辞書をデータベースに登録するには **pygeonlp add-dictionary** を使います。

```
$ pygeonlp add-dictionary mydic.json mydic.csv
```

必ず JSON ファイル名を先、 CSV ファイル名を後に指定してください。

登録できたことを確認するため、「東京タワー」をデータベースから検索してみます。

```
$ pygeonlp search 東京タワー
{"_4_mydic001": {"body": "東京タワー", "dictionary_id": 4, "entry_id": "mydic001", "geolod_
↪id": "_4_mydic001", "latitude": "35.65861", "longitude": "139.74556", "ne_class": "通信
施設", "url": "https://www.tokyotower.co.jp/", "dictionary_identifier": "geonlp:my-
↪buildings"}}
```

登録した地名語はジオパーズでも解析できます。

```
$ echo "東京タワーは今日も大人気。" | pygeonlp geoparse
東京タワー      名詞,固有名詞,地名語,_4_mydic001:東京タワー,*,*,東京タワー,,    通信施設,_4_
↪mydic001,東京タワー,139.74556,35.65861
は      助詞,係助詞,*,*,*,*,は,ハ,ワ
今日    名詞,副詞可能,*,*,*,*,今日,キョウ,キョー
も      助詞,係助詞,*,*,*,*,も,モ,モ
大人気  名詞,形容動詞語幹,*,*,*,*,大人気,ダイニンキ,ダイニンキ
。      記号,句点,*,*,*,*,。,。,。
EOS
```

第 10 章

カスタム解析モジュールの開発

ここでは、より高度な解析を行うモジュールを独自に開発したい場合に必要な情報を提供します。

10.1 デフォルトの解析フローをカスタマイズ

テキストを解析する *pygeonlp.api.geoparse()* は、実際には *pygeonlp.api.default_workflow()* が返すデフォルトの `Workflow` インスタンスを取得し、そのクラスメソッドである `geoparse()` を呼び出すショートカットです。そのため解析フローをカスタマイズするには、Workflow クラスの動作を理解する必要があります。

デフォルトの Workflow の **geoparse()** が実行する解析手順は次の通りです。

- *Parser* を利用して、テキストから**ラティス表現** を作成
- *Filter* を適用して、**ラティス表現** に含まれる候補を絞り込み、より限定された **ラティス表現** を作成
- *Evaluator* を利用して、**ラティス表現** から最もスコアの高い**パス表現** を作成
- **パス表現** を GeoJSON 形式に変換

Workflow クラスは parser, filters, evaluator の 3 つのメンバーを持ちます。

上記の手順をカスタマイズしたい場合、これらのメンバーを操作すれば処理を変更することができます。

単純な例として、市区町村クラスを持つ候補以外を削除する EntityClassFilter をセットするコードは以下のようになります。

```
import pygeonlp.api as api
from pygeonlp.api.filter import EntityClassFilter
api.init()
api.default_workflow().filters = [
    EntityClassFilter(r'市区町村/?.*')]
results = api.geoparse(text)
```

もう少し複雑な例として、Parser を独自拡張した MyParser に置き換えるには、parser を次のように上書きします。

```
import pygeonlp.api as api
api.init()
myparser = MyParser(my_parameter)
api.default_workflow().parser = myparser
results = api.geoparse(<text>)
```

10.2 独自ワークフロークラスを定義

デフォルト Workflow のメンバを書き換えるとその後の処理にも影響するため、想定しない副作用を生じる可能性があります。そのため、 Workflow クラスから派生した独自のワークフロークラス MyWorkflow を定義し、その **geoparse()** を呼び出す方が安全です。

```
from pygeonlp.api.workflow import Workflow

class MyWorkflow(Workflow):

    def __init__(self, my_parameter, **params):
        super().__init__(**params)
        self.parser = MyParser(my_parameter)

myworkflow = MyWorkflow(my_parameter)
results = myworkflow.geoparse(text)
```

もし parser, filters, evaluator を置き換えるだけでは実現できないロジックを実装したい場合、**Workflow.geoparse()** を参考にして独自ワークフロークラスの **geoparse()** メソッドを再定義してください。

10.3 実装サンプル

Filter, Evaluator, Workflow の拡張実装例が GitHub にありますので、参考にしてください。

第 11 章

pygeonlp の用語

ここでは pygeonlp で利用する用語を定義します。

11.1 地名語

GeoNLP ドキュメントの地名語を参照してください。

pygeonlp では、 **拡張辞書をインストール** するとその辞書に含まれる地名語がデータベースに登録され、地名解析に利用できるようになります。

11.2 地名解析辞書

GeoNLP ドキュメントの地名解析辞書を参照してください。

pygeonlp では、地名解析辞書は CSV 形式のファイルとして扱います。

11.3 データベース

pygeonlp では地名抽出を効率よく行なうため、地名語をデータベースに登録して管理しています。データベースには以下のファイルが含まれています。

- **geodic.sq3**

  地名解析辞書から読み込んだ地名語のリストを格納する SQLite3 データベースファイルです。

- **wordlist.sq3**

  地名語のリストから作成した、語の表記と geolod_id の関係を記録した SQLite3 データベースファイルです。

- **geo_name_fullname.drt**

テキスト中の地名語を効率よく抽出するには Common Prefix Search 処理を高速に行なう必要があります。pygeonlp では Darts を利用しており、登録されている全ての地名語表記から作成した Darts 辞書ファイルを **geo_name_fullname.drt** というファイル名で保存しています。

11.4 データベースディレクトリ

データベース を配置するディレクトリを指します。

データベースを複数用意したい場合、データベース内のファイル名は固定なので、ディレクトリを変更する必要があります（データベースはディレクトリごとコピーすれば複製できます）。

データベースディレクトリは次の順序で決定されます。

- API の **db_dir** パラメータで指定されたディレクトリ
- 環境変数 **GEONLP_DB_DIR** がセットされている場合はその値
- 環境変数 **HOME** がセットされている場合は **$HOME/geonlp/db/**

上記のいずれも当てはまらない場合は **RuntimeError** になります。

Docker や仮想環境で、コードを変更せずにデータベースディレクトリを変更したい場合には、環境変数 **HOME** または **GEONLP_DB_DIR** をセットするのが便利です。

一連の処理の中で、タスクによって利用するデータベースを切り替えたいという場合には、 **db_dir** パラメータにデータベースディレクトリを指定して *init()* を呼んでください。

```
python
>>> import pygeonlp.api as api
>>> api.init(db_dir='/usr/local/share/lib/geonlp/db')
>>> ... （共通データベースを利用したタスク） ...
>>> api.init(db_dir='/home/me/geonlp/testdb')
>>> ... （テスト用データベースを利用したタスク） ...
```

クラス *pygeonlp.api.workflow.Workflow* または *pygeonlp.api.service.Service* のコンストラクタで **db_dir** を指定することで、使用するデータベースを個別に指定することもできます。

11.5 ラティス表現

ラティス表現は、地名解析処理の途中で利用する内部データ表現です。

テキストを形態素に分解し、それぞれの形態素に対して 1 個以上の候補 *Node* オブジェクトが含まれる二重リスト構造になっています。

例： api.analyze("アメリカ大使館：港区赤坂 1-10-5") の結果として得られるラティス表現の構造のイメージ

```
[
  [ アメリカ大使館 ],
  [ ： ],
  [ 港区（市区町村・東京都），港区（市区町村・名古屋市），港区（市区町村・大阪市） ],
  [ 赤坂（駅・上毛電気鉄道／上毛線），赤坂（駅・東京メトロ／千代田線），
    赤坂（駅・富士急行／大月線），赤坂（駅・福岡市営地下鉄／空港線）],
  [ 1 ],
  [ - ],
  [ 10 ],
  [ - ],
  [ 5 ]
]
```

解析結果は 9 個の形態素からなり、3 番目の「港区」の形態素には 3 個の候補 Node が、4 番目の「赤坂」の形態素には 4 個の候補 Node があります。

ラティス表現は *pp_lattice()* を利用して簡易表示することができます。

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.devtool import pp_lattice
>>> lattice = api.analyze('アメリカ大使館：港区赤坂 1-10-5')
>>> pp_lattice(lattice)
#0:'アメリカ大使館'
  アメリカ大使館 (NORMAL)
#1:'：'
  ： (NORMAL)
#2:'港区'
  港区 (GEOWORD:['東京都'])
  港区 (GEOWORD:['愛知県', '名古屋市'])
  港区 (GEOWORD:['大阪府', '大阪市'])
#3:'赤坂'
  赤坂 (GEOWORD:['上毛電気鉄道', '上毛線'])
  赤坂 (GEOWORD:['東京地下鉄', '9 号線千代田線'])
```

(次のページに続く)

(前のページからの続き)

```
  赤坂(GEOWORD:['富士急行', '大月線'])
  赤坂(GEOWORD:['福岡市', '1号線(空港線)'])
#4:'1'
  1(NORMAL)
#5:'-'
  -(NORMAL)
#6:'10'
  10(NORMAL)
#7:'-'
  -(NORMAL)
#8:'5'
  5(NORMAL)
```

住所解析を行なうと、住所候補を構成する形態素に含まれる「住所以外の候補」は削除され、住所ノードに統合されます。

例： api.analyze("アメリカ大使館：港区赤坂 1-10-5", jageocoder=True)

```
#0:'アメリカ大使館'
  アメリカ大使館(NORMAL)
#1:'：'
  ：(NORMAL)
#2:'港区赤坂1-10-'
  港区赤坂1-10-(ADDRESS:東京都/港区/赤坂/一丁目/10番)[6]
#3:'5'
  5(NORMAL)
```

住所以外の候補も残したい場合は **keep_nodes=True** を指定します。この場合、住所に該当する先頭の形態素に住所ノードが追加されます。

例： api.analyze("アメリカ大使館：港区赤坂 1-10-5", jageocoder=True, keep_nodes=True)

```
#0:'アメリカ大使館'
  アメリカ大使館(NORMAL)
#1:'：'
  ：(NORMAL)
#2:'港区'
  港区(GEOWORD:['東京都'])
  港区(GEOWORD:['愛知県', '名古屋市'])
  港区(GEOWORD:['大阪府', '大阪市'])
```

(次のページに続く)

(前のページからの続き)

```
  港区赤坂 1-10-(ADDRESS: 東京都/港区/赤坂/一丁目/10 番)[6]
#3:' 赤坂'
  赤坂 (GEOWORD:[' 上毛電気鉄道', ' 上毛線'])
  赤坂 (GEOWORD:[' 東京地下鉄', '9 号線千代田線'])
  赤坂 (GEOWORD:[' 富士急行', ' 大月線'])
  赤坂 (GEOWORD:[' 福岡市', '1 号線 (空港線)'])
#4:'1'
  1(NORMAL)
#5:'-'
  -(NORMAL)
#6:'10'
  10(NORMAL)
#7:'-'
  -(NORMAL)
#8:'5'
  5(NORMAL)
```

11.6 パス表現

パス表現は、一連の地名解析処理のうち、最後のスコアリングと結果の出力の際に利用する内部データ表現です。

テキストを形態素に分解し、それぞれの形態素に対する候補から 1 つずつ選択した *Node* オブジェクトのリスト構造になっています。

LinkedResults ジェネレータクラスを利用すると、ラティス表現からパス表現の候補を一つずつ生成して取得することができます。

例： next(LinkedResults(api.analyze(' アメリカ大使館：港区赤坂 1-10-5'))) の結果として得られるパス表現の構造のイメージ

```
[
  アメリカ大使館,
  ：,
  港区（市区町村・東京都）,
  赤坂（駅・上毛電気鉄道／上毛線）,
  1,
  -,
  10,
  -,
```

(次のページに続く)

(前のページからの続き)

```
  5
]
```

このセンテンスを解析すると「港区」の候補が 3 個、「赤坂」の候補が 4 個存在するため、 3 × 4 = 12 個のパス表現が得られます。

パス表現は *pp_path()* を利用して簡易表示することができます。以降の例ではこの簡易表示を利用します。

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.linker import LinkedResults
>>> from pygeonlp.api.devtool import pp_path
>>> api.init()
>>> lattice = api.analyze('アメリカ大使館：港区赤坂1-10-5')
>>> pp_path(next(LinkedResults(lattice)))
[
  #0:アメリカ大使館(NORMAL)
  #1:：(NORMAL)
  #2:港区(GEOWORD:['東京都'])
  #3:赤坂(GEOWORD:['上毛電気鉄道', '上毛線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
```

住所ノードを含むラティス表現からパス表現を生成する場合、住所ノードが複数の形態素にまたがるため、次のノードを正しく選択する必要があります。

LinkedResults はこの処理を自動的に行ないます。

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.linker import LinkedResults
>>> from pygeonlp.api.devtool import pp_lattice, pp_path
>>> lattice = api.analyze('アメリカ大使館：港区赤坂1-10-5', jageocoder=True, keep_
↪nodes=True)
>>> for path in LinkedResults(lattice):
...   pp_path(path)
...
[
```

(次のページに続く)

(前のページからの続き)

```
  #0: アメリカ大使館 (NORMAL)
  #1: : (NORMAL)
  #2: 港区 (GEOWORD:[' 東京都'])
  #3: 赤坂 (GEOWORD:[' 上毛電気鉄道', ' 上毛線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0: アメリカ大使館 (NORMAL)
  #1: : (NORMAL)
  #2: 港区 (GEOWORD:[' 東京都'])
  #3: 赤坂 (GEOWORD:[' 東京地下鉄', '9 号線千代田線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0: アメリカ大使館 (NORMAL)
  #1: : (NORMAL)
  #2: 港区 (GEOWORD:[' 東京都'])
  #3: 赤坂 (GEOWORD:[' 富士急行', ' 大月線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0: アメリカ大使館 (NORMAL)
  #1: : (NORMAL)
  #2: 港区 (GEOWORD:[' 東京都'])
  #3: 赤坂 (GEOWORD:[' 福岡市', '1 号線 (空港線)'])
  #4:1(NORMAL)
  #5:-(NORMAL)
```

(次のページに続く)

(前のページからの続き)

```
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0:アメリカ大使館(NORMAL)
  #1:：(NORMAL)
  #2:港区(GEOWORD:['愛知県', '名古屋市'])
  #3:赤坂(GEOWORD:['上毛電気鉄道', '上毛線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0:アメリカ大使館(NORMAL)
  #1:：(NORMAL)
  #2:港区(GEOWORD:['愛知県', '名古屋市'])
  #3:赤坂(GEOWORD:['東京地下鉄', '9号線千代田線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0:アメリカ大使館(NORMAL)
  #1:：(NORMAL)
  #2:港区(GEOWORD:['愛知県', '名古屋市'])
  #3:赤坂(GEOWORD:['富士急行', '大月線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0:アメリカ大使館(NORMAL)
```

(次のページに続く)

(前のページからの続き)

```
  #1:：(NORMAL)
  #2: 港区 (GEOWORD:[' 愛知県', ' 名古屋市'])
  #3: 赤坂 (GEOWORD:[' 福岡市', '1 号線 (空港線)'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0: アメリカ大使館 (NORMAL)
  #1:：(NORMAL)
  #2: 港区 (GEOWORD:[' 大阪府', ' 大阪市'])
  #3: 赤坂 (GEOWORD:[' 上毛電気鉄道', ' 上毛線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0: アメリカ大使館 (NORMAL)
  #1:：(NORMAL)
  #2: 港区 (GEOWORD:[' 大阪府', ' 大阪市'])
  #3: 赤坂 (GEOWORD:[' 東京地下鉄', '9 号線千代田線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0: アメリカ大使館 (NORMAL)
  #1:：(NORMAL)
  #2: 港区 (GEOWORD:[' 大阪府', ' 大阪市'])
  #3: 赤坂 (GEOWORD:[' 富士急行', ' 大月線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
```

(次のページに続く)

(前のページからの続き)

```
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0: アメリカ大使館 (NORMAL)
  #1:：(NORMAL)
  #2: 港区 (GEOWORD:[' 大阪府', ' 大阪市'])
  #3: 赤坂 (GEOWORD:[' 福岡市', '1 号線 (空港線)'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0: アメリカ大使館 (NORMAL)
  #1:：(NORMAL)
  #2: 港区赤坂 1-10-(ADDRESS: 東京都/港区/赤坂/一丁目/10 番)[6]
  #3:5(NORMAL)
]
```

pygeonlp の地名解決処理では、パス表現ごとのスコアを *path_score()* で計算し、降順にソートして結果を返します。

パス表現のスコア計算方法をカスタマイズしたい場合は**スコアリング方法のカスタマイズ** を参照してください。

第 12 章

JSON 表現

pygeonlp では、 JSON を利用して地名語 や住所の情報を表現します。

ここではこれらの情報の JSON 表現について説明します。

12.1 地名語の JSON 表現

地名語の JSON 表記例を示します。 GeoJSON に準拠しています。

- properties.node_type は *GEOWORD*
- properties.morphemes に MeCab の処理結果を格納
- properties.geoword_properties に地名解析辞書の情報を格納

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [
      139.757845,
      35.6960275
    ]
  },
  "properties": {
    "surface": "神保町",
    "node_type": "GEOWORD",
    "morphemes": {
      "conjugated_form": "*",
      "conjugation_type": "*",
```

(次のページに続く)

(前のページからの続き)


```
      "original_form": "神保町",
      "pos": "名詞",
      "prononciation": "",
      "subclass1": "固有名詞",
      "subclass2": "地名語",
      "subclass3": "82wiE0: 神保町駅",
      "surface": "神保町",
      "yomi": ""
    },
    "geoword_properties": {
      "body": "神保町",
      "dictionary_id": 4,
      "entry_id": "2891e10e9314a0b378fac6aace6d2a7f",
      "geolod_id": "82wiE0",
      "hypernym": [
        "東京都",
        "10 号線新宿線"
      ],
      "institution_type": "公営鉄道",
      "latitude": "35.6960275",
      "longitude": "139.757845",
      "ne_class": "鉄道施設/鉄道駅",
      "railway_class": "普通鉄道",
      "suffix": [
        "駅",
        ""
      ],
      "dictionary_identifier": "geonlp:ksj-station-N02-2019"
    }
  }
}
```

12.2 住所の JSON 表現

住所の JSON 表記例を示します。 GeoJSON に準拠しています。

- properties.node_type は *ADDRESS*
- properties.morphemes に住所文字列を構成するそれぞれの単語の情報を格納
- properties.address_properties にジオコーダーの解析結果を格納

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [
      139.758148,
      35.692332
    ]
  },
  "properties": {
    "surface": "千代田区一ツ橋２－１－",
    "node_type": "ADDRESS",
    "morphemes": [{
        "surface": "千代田区",
        "node_type": "GEOWORD",
        "morphemes": {
          "conjugated_form": "*",
          "conjugation_type": "*",
          "original_form": "千代田区",
          "pos": "名詞",
          "prononciation": "",
          "subclass1": "固有名詞",
          "subclass2": "地名語",
          "subclass3": "WWIY7G:千代田区",
          "surface": "千代田区",
          "yomi": ""
        },
        "geometry": {
          "type": "Point",
          "coordinates": [
            139.753634,
            35.694003
```

(次のページに続く)

(前のページからの続き)


```
        ]
      },
      "prop": {
        "address": "東京都千代田区",
        "body": "千代田",
        "body_variants": "千代田",
        "code": {},
        "countyname": "",
        "countyname_variants": "",
        "dictionary_id": 1,
        "entry_id": "13101A1968",
        "geolod_id": "WWIY7G",
        "hypernym": [
          "東京都"
        ],
        "latitude": "35.69400300",
        "longitude": "139.75363400",
        "ne_class": "市区町村",
        "prefname": "東京都",
        "prefname_variants": "東京都",
        "source": "1/千代田区役所/千代田区九段南 1-2-1/P34-14_13.xml",
        "suffix": [
          "区"
        ],
        "valid_from": "",
        "valid_to": "",
        "dictionary_identifier": "geonlp:geoshape-city"
      }
    },
    {
      "surface": "一ツ橋",
      "node_type": "NORMAL",
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "一ツ橋",
        "pos": "名詞",
        "prononciation": "ヒトツバシ",
        "subclass1": "固有名詞",
```


(次のページに続く)

(前のページからの続き)


```
        "subclass2": "地域",
        "subclass3": "一般",
        "surface": "一ツ橋",
        "yomi": "ヒトツバシ"
      },
      "geometry": null,
      "prop": null
    },
    {
      "surface": "２",
      "node_type": "NORMAL",
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "２",
        "pos": "名詞",
        "prononciation": "ニ",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "２",
        "yomi": "ニ"
      },
      "geometry": null,
      "prop": null
    },
    {
      "surface": "ー",
      "node_type": "NORMAL",
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "記号",
        "prononciation": "",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "ー",
```


(次のページに続く)

(前のページからの続き)


```
        "yomi": ""
      },
      "geometry": null,
      "prop": null
    },
    {
      "surface": "1",
      "node_type": "NORMAL",
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "1",
        "pos": "名詞",
        "prononciation": "イチ",
        "subclass1": "数",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "1",
        "yomi": "イチ"
      },
      "geometry": null,
      "prop": null
    },
    {
      "surface": "ー",
      "node_type": "NORMAL",
      "morphemes": {
        "conjugated_form": "*",
        "conjugation_type": "*",
        "original_form": "*",
        "pos": "記号",
        "prononciation": "",
        "subclass1": "一般",
        "subclass2": "*",
        "subclass3": "*",
        "surface": "ー",
        "yomi": ""
      },
      "geometry": null,
```


(次のページに続く)

(前のページからの続き)

```
        "prop": null
      }
    ],
    "address_properties": {
      "id": 11460296,
      "name": "1 番",
      "x": 139.758148,
      "y": 35.692332,
      "level": 7,
      "note": null,
      "fullname": [
        "東京都",
        "千代田区",
        "一ツ橋",
        "二丁目",
        "1 番"
      ]
    }
  }
}
```

12.3 地名解析辞書メタデータの JSON 表現

地名解析辞書メタデータの JSON 表記例を示します。

- 都道府県辞書の例

```
{
  "@context": "https://schema.org/",
  "@type": "Dataset",
  "alternateName": "",
  "creator": [{
    "@type": "Organization",
    "name": "GeoNLP Project",
    "sameAs": "http://geonlp.ex.nii.ac.jp/"
  }],
  "dateModified": "2014-01-15T14:36:17+09:00",
  "description": "1 都 1 道 2 府 43 県の辞書です。県庁所在地の住所を含みます。自由フィールドとして県庁
```

(次のページに続く)

(前のページからの続き)

```
の代表電話番号も含みます。代表点の座標は県庁の住所をジオコーディングにより取得しています。",
  "distribution": [{
    "@type": "DataDownload",
    "contentUrl": "http://agora.ex.nii.ac.jp/GeoNLP/dict/geoshape-pref.csv",
    "encodingFormat": "text/csv"
  }],
  "identifier": ["geonlp:geoshape-pref"],
  "isBasedOn": {
    "@type": "CreativeWork",
    "name": "財団法人地方自治情報センター「都道府県庁」",
    "url": "https://www.lasdec.or.jp/cms/1,69,14,188,html"
  },
  "keywords": ["GeoNLP", "地名辞書"],
  "license": "https://creativecommons.org/licenses/by/4.0/",
  "name": "日本の都道府県（2010 年 4 月）",
  "size": "48",
  "spatialCoverage": {
    "@type": "Place",
    "geo": {
      "@type": "GeoShape",
      "box": "26.2133 126.67963 43.06411 141.34702"
    }
  },
  "temporalCoverage": "../..",
  "url": "https://geonlp.ex.nii.ac.jp/dictionary/geoshape-pref/"
}
```

- 市区町村辞書の例

```
{
  "@context": "https://schema.org/",
  "@type": "Dataset",
  "alternateName": "",
  "creator": [{
    "@type": "Organization",
    "name": "ROIS-DS 人文学オープンデータ共同利用センター",
    "sameAs": "http://codh.rois.ac.jp/"
  }],
  "dateModified": "2021-01-04T22:03:51+09:00",
```

(次のページに続く)

(前のページからの続き)


```
  "description": "歴史的行政区域データセットβ版で構築した地名辞書です。1920 年から 2020 年までの国土数値情報「行政区域データ」に出現する市区町村をリスト化し、独自の固有 ID を付与して公開しています。データセット構築の詳しい手法については、「歴史的行政区域データセットβ版」のウェブサイトをご覧ください。",
  "distribution": [{
    "@type": "DataDownload",
    "contentUrl": "http://agora.ex.nii.ac.jp/GeoNLP/dict/geoshape-city.csv",
    "encodingFormat": "text/csv"
  }],
  "identifier": ["geonlp:geoshape-city"],
  "isBasedOn": {
    "@type": "CreativeWork",
    "name": "歴史的行政区域データセットβ版",
    "url": "https://geoshape.ex.nii.ac.jp/city/"
  },
  "keywords": ["GeoNLP", "地名辞書"],
  "license": "https://creativecommons.org/licenses/by/4.0/",
  "name": "歴史的行政区域データセットβ版地名辞書",
  "size": "16421",
  "spatialCoverage": {
    "@type": "Place",
    "geo": {
      "@type": "GeoShape",
      "box": "24.06092 123.004496 45.5566280626738 148.772556996888"
    }
  },
  "temporalCoverage": "../..",
  "url": "https://geonlp.ex.nii.ac.jp/dictionary/geoshape-city/"
}
```

第 13 章

環境変数

pygeonlp で利用できる環境変数は以下の通りです。

GEONLP_DB_DIR

データベースディレクトリ を指定します。

指定しない場合、 **$(HOME)/geonlp/db** にデータベースを作成します。HOME も設定されていない場合はエラーになります。

GEONLP_MAX_COMBINATIONS 解

析可能なパス表現の最大値を指定します。文を解析した時に、たとえば 3 番目の単語に 4 種類の候補があり、6 番目の単語に 3 種類の候補が、8 番目の単語に 5 種類の候補があると、パス表現の数はこれらの組み合わせなので 4 × 3 × 5=60 になります。

組み合わせの数が最大値を超えると `LinkerError` になります。

指定しない場合は 256 です。もし **LinkerError** が出てしまう場合は文をより短く分解するか、**GEONLP_MAX_COMBINATIONS** の値を大きめに指定してください。

GEONLP_EXCLUDED_WORD 地

名語として解析したくない語を指定します。複数の語を指定したい場合は **|** を区切りとして列挙してください。詳しくは **特定の語を抽出しない** を参照してください。

指定しない場合は **本部|一部|月** です。

GEONLP_ADDRESS_CLASS 住

所の先頭として抽出する地名語の固有名クラスを正規表現で指定します。

たとえば「NII は千代田区一ツ橋にあります」という文から「千代田区一ツ橋」を住所として抽出するには、「千代田区」の固有名クラスである「市区町村」が **GEONLP_ADDRESS_CLASS** に指定されている正規表現とマッチする必要があります。

指定しない場合は **r'^(都道府県|市区町村|行政地域|居住地名)(/.+|)'** です。

GEONLP_MECAB_DIC_DIR 形

態素解析に利用する MeCab システム辞書のディレクトリを指定します。

NEologd **連携** を利用する場合は NEologd の辞書をインストールしたディレクトリを指定してください。

指定しない場合は MeCab のシステム辞書を利用します。

JAGEOCODER_DB2_DIR 住

所ジオコーダ jageocoder の住所辞書をインストールしたディレクトリを指定します。

指定しない場合は Python ライブラリがインストールされるディレクトリの下の **jageocoder/db2** にインストールされます。

jageocoder get-db-dir コマンドで確認できます。

第 14 章

pygeonlp.api package

pygeonlp.api パッケージを構成するサブモジュール、およびメソッドのリファレンス。

14.1 API モジュール

pygeonlp.api.**activateDictionaries**(*idlist=None*, *pattern=None*)

指定した辞書を再び利用するようにします。既に利用可能な辞書は指定しなくても利用可能なままになります。

パラメータ

- **idlist** (*list*) -- 利用する辞書の内部 id または identifier のリスト。
- **pattern** (*str*) -- 利用する辞書の identifier を指定する正規表現。

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> sorted(api.disactivateDictionaries(pattern=r'.*'))
['geonlp:geoshape-city', 'geonlp:geoshape-pref', 'geonlp:ksj-station-N02']
>>> sorted([x.get_identifier() for x in api.getActiveDictionaries()])
[]
>>> api.activateDictionaries(pattern=r'ksj-station')
['geonlp:ksj-station-N02']
>>> sorted([x.get_identifier() for x in api.getActiveDictionaries()])
['geonlp:ksj-station-N02']
```

注釈: idlist と pattern を同時に指定した場合、どちらか一方の条件に一致する辞書は利用可能になります。

pygeonlp.api.**addDictionaryFromFile**(*jsonfile*, *csvfile*)

指定したパスにある辞書メタデータ（JSON ファイル）と地名解析辞書（CSV ファイル）をデータベースに登録します。

既に同じ identifier を持つ辞書データがデータベースに登録されている場合、削除してから新しい辞書データを登録します。

登録した辞書を利用可能にするには、 `setActivateDictionaries()` または `activateDictionaires()` で有効化する必要があります。

パラメータ

- **jsonfile** (*str*) -- 辞書メタデータファイルのパス。
- **csvfile** (*str*) -- 地名解析辞書ファイルのパス。

戻り値

常に True。登録に失敗した場合は例外が発生します。

戻り値の型

bool

pygeonlp.api.**addDictionaryFromWeb**(*url*, *params=None*, ***kwargs*)

指定した URL にあるページに含まれる辞書メタデータ（JSON-LD）を取得し、メタデータに記載されている URL から地名解析辞書（CSV ファイル）を取得し、データベースに登録します。

既に同じ identifier を持つ辞書データがデータベースに登録されている場合、削除してから新しい辞書データを登録します。

登録した辞書を利用可能にするには、 `setActivateDictionaries()` または `activateDictionaires()` で有効化する必要があります。

パラメータ

- **url** (*str*) -- 辞書メタデータを含むウェブページの URL。
- **params** (*dict, optional*) -- requests.get に渡す params パラメータ。
- ****kwargs** (*dict, optional*) -- requests.get に渡す kwargs パラメータ。

戻り値

常に True。登録に失敗した場合は例外が発生します。

戻り値の型

bool

pygeonlp.api.**analyze**(*sentence*, ***kwargs*)

文を解析した結果をラティス表現で返します。

パラメータ

sentence (*str*) -- 解析するテキスト。

戻り値

解析結果のラティス表現。

戻り値の型

list

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.analyze('今日は国会議事堂前まで歩きました。')
[[{"surface": "今日", "node_type": "NORMAL", "morphemes": {"conjugated_form": "*",
↪"conjugation_type": "*", "original_form": "今日", "pos": "名詞", "prononciation":
↪"キョー", "subclass1": "副詞可能", "subclass2": "*", "subclass3": "*", "surface":
↪"今日", "yomi": "キョウ"}, "geometry": null, "prop": null}], [{"surface": "は",
 "node_type": "NORMAL", "morphemes": {"conjugated_form": "*", "conjugation_type": "*
↪", "original_form": "は", "pos": "助詞", "prononciation": "ワ", "subclass1": "係助詞
↪", "subclass2": "*", "subclass3": "*", "surface": "は", "yomi": "ハ"}, "geometry":␣
↪null, "prop": null}], [{"surface": "国会議事堂前", "node_type": "GEOWORD",
 "morphemes": {"conjugated_form": "*", "conjugation_type": "*", "original_form": "国
会議事堂前", "pos": "名詞", "prononciation": "", "subclass1": "固有名詞", "subclass2":
↪"地名語", "subclass3": "Bn4q6d: 国会議事堂前駅", "surface": "国会議事堂前", "yomi": "
↪"}, "geometry": {"type": "Point", "coordinates": [139.74534166666666, 35.674845]},
 "prop": {"body": "国会議事堂前", "dictionary_id": 3, "entry_id": "LrGGxY", "geolod_
↪id": "Bn4q6d", "hypernym": ["東京地下鉄", "4 号線丸ノ内線"], "institution_type": "民営
鉄道", "latitude": "35.674845", "longitude": "139.74534166666666", "ne_class": "鉄道
施設/鉄道駅", "railway_class": "普通鉄道", "suffix": ["駅", ""], "dictionary_
↪identifier": "geonlp:ksj-station-N02"}}, {"surface": "国会議事堂前", "node_type":
↪"GEOWORD", "morphemes": {"conjugated_form": "*", "conjugation_type": "*",
 "original_form": "国会議事堂前", "pos": "名詞", "prononciation": "", "subclass1": "固
有名詞", "subclass2": "地名語", "subclass3": "cE8W4w: 国会議事堂前駅", "surface": "国会
議事堂前", "yomi": ""}, "geometry": {"type": "Point", "coordinates": [139.
↪74305333333334, 35.673543333333335]}, "prop": {"body": "国会議事堂前", "dictionary_
↪id": 3, "entry_id": "4NFELa", "geolod_id": "cE8W4w", "hypernym": ["東京地下鉄", "9
号線千代田線"], "institution_type": "民営鉄道", "latitude": "35.673543333333335",
```

(次のページに続く)

(前のページからの続き)


```
↪"longitude": "139.74305333333334", "ne_class": "鉄道施設/鉄道駅", "railway_class":
↪"普通鉄道", "suffix": ["駅", ""], "dictionary_identifier": "geonlp:ksj-station-N02
↪"}}], [{"surface": "まで", "node_type": "NORMAL", "morphemes": {"conjugated_form":
↪"*", "conjugation_type": "*", "original_form": "まで", "pos": "助詞",
 "prononciation": "マデ", "subclass1": "副助詞", "subclass2": "*", "subclass3": "*",
↪"surface": "まで", "yomi": "マデ"}, "geometry": null, "prop": null}], [{"surface":
↪"歩き", "node_type": "NORMAL", "morphemes": {"conjugated_form": "五段・カ行イ音便",
↪"conjugation_type": "連用形", "original_form": "歩く", "pos": "動詞", "prononciation
↪": "アルキ", "subclass1": "自立", "subclass2": "*", "subclass3": "*", "surface":
↪"歩き", "yomi": "アルキ"}, "geometry": null, "prop": null}], [{"surface": "まし",
↪"node_type": "NORMAL", "morphemes": {"conjugated_form": "特殊・マス", "conjugation_
↪type": "連用形", "original_form": "ます", "pos": "助動詞", "prononciation": "マシ",
 "subclass1": "*", "subclass2": "*", "subclass3": "*", "surface": "まし", "yomi": "マ
シ"}, "geometry": null, "prop": null}], [{"surface": "た", "node_type": "NORMAL",
 "morphemes": {"conjugated_form": "特殊・タ", "conjugation_type": "基本形", "original_
↪form": "た", "pos": "助動詞", "prononciation": "タ", "subclass1": "*", "subclass2
↪": "*", "subclass3": "*", "surface": "た", "yomi": "タ"}, "geometry": null, "prop
↪": null}], [{"surface": "。", "node_type": "NORMAL", "morphemes": {"conjugated_
↪form": "*", "conjugation_type": "*", "original_form": "。", "pos": "記号",
↪"prononciation": "。", "subclass1": "句点", "subclass2": "*", "subclass3": "*",
↪"surface": "。", "yomi": "。"}, "geometry": null, "prop": null}]]
```

注釈: ラティス表現では全ての地名語の候補を列挙して返します。

---

pygeonlp.api.**clearDatabase**()

地名語データベースに登録されている辞書をクリアします。データベース内の地名語も全て削除されます。

この関数は、データベースを作り直す際に利用します。

pygeonlp.api.**default_workflow**()

Default Workflow オブジェクトを返します。

通常はこのメソッドを利用する必要はありません。

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> type(api.default_workflow())
<class 'pygeonlp.api.workflow.Workflow'>
```

pygeonlp.api.**disactivateDictionaries**(*idlist=None*, *pattern=None*)

指定した辞書を一時的に除外し、利用しないようにします。既に除外されている辞書は除外されたままになります。

パラメータ

- **idlist** (*list*) -- 除外する辞書の内部 id または identifier のリスト。
- **pattern** (*str*) -- 除外する辞書の identifier を指定する正規表現。

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> sorted(api.disactivateDictionaries(pattern=r'geonlp:geoshape'))
['geonlp:geoshape-city', 'geonlp:geoshape-pref']
>>> [x.get_identifier() for x in api.getActiveDictionaries()]
['geonlp:ksj-station-N02']
>>> api.disactivateDictionaries(pattern=r'ksj-station')
['geonlp:ksj-station-N02']
>>> [x.get_identifier() for x in api.getActiveDictionaries()]
[]
```

注釈: idlist と pattern を同時に指定した場合、どちらか一方の条件に一致する辞書は除外されます。

---

pygeonlp.api.**geoparse**(*sentence*)

Default Workflow を利用して文を解析した結果を、GeoJSON Feature 形式に変換可能な dict のリストとして返します。

パラメータ

sentence (*str*) -- 解析する文字列。

戻り値

形態素で分割した各要素のジオメトリや形態素情報、地名語情報を GeoJSON Feature 形式に変換可

能な dict で表現し、そのリストを返します。

戻り値の型

list

サンプル

```
>>> import json
>>> import pygeonlp.api as api
>>> api.init()
>>> json.dumps({'type':'FeatureCollection', 'features':api.geoparse('今日は国会議事堂
前まで歩きました。')}, ensure_ascii=False)
'{"type": "FeatureCollection", "features": [{"type": "Feature", "geometry": null,
↪"properties": {"surface": "今日", "node_type": "NORMAL", "morphemes": {"conjugated_
↪form": "*", "conjugation_type": "*", "original_form": "今日", "pos": "名詞",
 "prononciation": "キョー", "subclass1": "副詞可能", "subclass2": "*", "subclass3": "*
↪", "surface": "今日", "yomi": "キョウ"}}}, {"type": "Feature", "geometry": null,
↪"properties": {"surface": "は", "node_type": "NORMAL", "morphemes": {"conjugated_
↪form": "*", "conjugation_type": "*", "original_form": "は", "pos": "助詞",
↪"prononciation": "ワ", "subclass1": "係助詞", "subclass2": "*", "subclass3": "*",
 "surface": "は", "yomi": "ハ"}}}, {"type": "Feature", "geometry": {"type": "Point",
↪"coordinates": [139.74305333333334, 35.673543333333335]}, "properties": {"surface
↪": "国会議事堂前", "node_type": "GEOWORD", "morphemes": {"conjugated_form": "*",
↪"conjugation_type": "*", "original_form": "国会議事堂前", "pos": "名詞",
↪"prononciation": "", "subclass1": "固有名詞", "subclass2": "地名語", "subclass3":
↪"cE8W4w:国会議事堂前駅", "surface": "国会議事堂前", "yomi": ""}, "geoword_properties
↪": {"body": "国会議事堂前", "dictionary_id": 3, "entry_id": "4NFELa", "geolod_id":
↪"cE8W4w", "hypernym": ["東京地下鉄", "9号線千代田線"], "institution_type": "民営鉄道
", "latitude": "35.673543333333335", "longitude": "139.74305333333334", "ne_class":
↪"鉄道施設/鉄道駅", "railway_class": "普通鉄道", "suffix": ["駅", ""], "dictionary_
↪identifier": "geonlp:ksj-station-N02"}}}, {"type": "Feature", "geometry": null,
↪"properties": {"surface": "まで", "node_type": "NORMAL", "morphemes": {"conjugated_
↪form": "*", "conjugation_type": "*", "original_form": "まで", "pos": "助詞",
 "prononciation": "マデ", "subclass1": "副助詞", "subclass2": "*", "subclass3": "*",
↪"surface": "まで", "yomi": "マデ"}}}, {"type": "Feature", "geometry": null,
↪"properties": {"surface": "歩き", "node_type": "NORMAL", "morphemes": {"conjugated_
↪form": "五段・カ行イ音便", "conjugation_type": "連用形", "original_form": "歩く",
↪"pos": "動詞", "prononciation": "アルキ", "subclass1": "自立", "subclass2": "*",
↪"subclass3": "*", "surface": "歩き", "yomi": "アルキ"}}}, {"type": "Feature",
↪"geometry": null, "properties": {"surface": "まし", "node_type": "NORMAL",
```

(次のページに続く)

(前のページからの続き)

```
↪"morphemes": {"conjugated_form": "特殊・マス", "conjugation_type": "連用形",
 "original_form": "ます", "pos": "助動詞", "prononciation": "マシ", "subclass1": "*",
↪"subclass2": "*", "subclass3": "*", "surface": "まし", "yomi": "マシ"}}}, {"type":
↪"Feature", "geometry": null, "properties": {"surface": "た", "node_type": "NORMAL
↪", "morphemes": {"conjugated_form": "特殊・タ", "conjugation_type": "基本形",
↪"original_form": "た", "pos": "助動詞", "prononciation": "タ", "subclass1": "*",
↪"subclass2": "*", "subclass3": "*", "surface": "た", "yomi": "タ"}}}, {"type":
↪"Feature", "geometry": null, "properties": {"surface": "。", "node_type": "NORMAL
↪", "morphemes": {"conjugated_form": "*", "conjugation_type": "*", "original_form
↪": "。", "pos": "記号", "prononciation": "。", "subclass1": "句点", "subclass2": "*
↪", "subclass3": "*", "surface": "。", "yomi": "。"}}}]}'
```

注釈: このメソッドは、文字列の解析、フィルタによる絞り込み、ランキングなどの一連の処理を行ないます。

pygeonlp.api.**getActiveClasses**()

解析に利用する固有名クラスの一覧を返します。デフォルトは '.*' で、全ての固有名クラスが利用されます。

一時的に特定の固有名クラスだけを解析対象としたい場合、setActiveClasses() で対象クラスを指定できます。

戻り値

利用する固有名クラス（正規表現）のリスト。

戻り値の型

list

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.getActiveClasses()
['.*']
```

pygeonlp.api.**getActiveDictionaries**()

インストール済み辞書のうち、解析に利用する辞書のメタデータ一覧を返します。デフォルトでは全てのインストール済み辞書を利用します。

一時的に辞書を利用したくない場合、 disactivateDictionaries() で除外できます。除外された辞書は

activateDictionaries() で再び利用可能になります。

戻り値

Metadata インスタンスのリスト。

戻り値の型

list

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> sorted([x.get_identifier() for x in api.getActiveDictionaries()])
['geonlp:geoshape-city', 'geonlp:geoshape-pref', 'geonlp:ksj-station-N02']
```

pygeonlp.api.**getDictionaries**()

インストール済み辞書のメタデータ一覧を返します。

戻り値

Metadata インスタンスのリスト。

戻り値の型

list

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> sorted([x.get_identifier() for x in api.getDictionaries()])
['geonlp:geoshape-city', 'geonlp:geoshape-pref', 'geonlp:ksj-station-N02']
```

pygeonlp.api.**getDictionary**(*id_or_identifier*)

指定した id または identifier を持つ辞書のメタデータを返します。id, identifier に一致する辞書が存在しない場合は None を返します。

パラメータ

id_or_identifier (*str or int*) -- str の場合は辞書 identifier で指定。int の場合は内部辞書 id で指定。

戻り値

Metadata インスタンス。

戻り値の型

Metadata

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.getDictionary('geonlp:ksj-station-N02')
{"@context": "https://schema.org/", "@type": "Dataset", "alternateName": "",
↪"creator": [{"@type": "Organization", "name": "株式会社情報試作室", "sameAs":
↪"https://www.info-proto.com/"}], "dateModified": "2021-08-27T17:18:18+09:00",
↪"description": "国土数値情報「鉄道データ（N02)」から作成した、日本の鉄道駅（地下鉄を含む）の
辞書です。hypernym には運営者名と路線名を記載しています。「都営」ではなく「東京都」のようになっ
ていますので注意してください。自由フィールドとして、railway_class に「鉄道区分」、institution_
↪type に「事業者種別」を含みます。", "distribution": [{"@type": "DataDownload",
↪"contentUrl": "https://www.info-proto.com/static/ksj-station-N02.csv",
↪"encodingFormat": "text/csv"}], "identifier": ["geonlp:ksj-station-N02"],
↪"isBasedOn": {"@type": "CreativeWork", "name": "国土数値情報 鉄道データ", "url":
↪"https://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-N02-v2_2.html"}, "keywords": [
↪"GeoNLP", "地名辞書"], "license": "https://creativecommons.org/licenses/by/4.0/",
 "name": "国土数値情報 鉄道データ（駅）", "size": "10191", "spatialCoverage": {"@type":
↪"Place", "geo": {"@type": "GeoShape", "box": "26.193265 127.652285 45.
↪41616333333333 145.59723"}}, "temporalCoverage": "../..", "url": "https://www.
↪info-proto.com/static/ksj-station-N02.html"}
```

pygeonlp.api.**getWordInfo**(*geolod_id*)

指定した geolod_id を持つ語の情報を返します。id が辞書に存在しない場合は None を返します。*deprecated* この関数は低水準 API なので、廃止予定です。

パラメータ

geolod_id (*str or Iterable*) -- 語の ID または ID のリストを返すイテレータ。

戻り値

geolod_id をキー、語の情報を値に持つ dict。

戻り値の型

dict

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.getWordInfo('Bn4q6d')
{'body': '国会議事堂前', 'dictionary_id': 3, 'entry_id': 'LrGGxY', 'geolod_id':
↪'Bn4q6d', 'hypernym': ['東京地下鉄', '4 号線丸ノ内線'], 'institution_type': '民営鉄道
', 'latitude': '35.674845', 'longitude': '139.74534166666666', 'ne_class': '鉄道施設/
↪鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅', ''], 'dictionary_identifier
↪': 'geonlp:ksj-station-N02'}
```

pygeonlp.api.**get_db_dir**()

データベースディレクトリを取得します。データベースディレクトリは次の優先順位に従って決定します。

- 環境変数 GEONLP_DB_DIR の値
- 環境変数 HOME が指すディレクトリの下の geonlp/db/

どちらの環境変数も指定されていない場合は RuntimeError を送出して終了します。

戻り値

ディレクトリの絶対パス。

戻り値の型

str

pygeonlp.api.**get_jageocoder_db_dir**()

deprecated

jageocoder の辞書が配置されているディレクトリを取得します。次の優先順位に従って決定します。

- 環境変数 JAGEOCODER_DB_DIR の値
- 環境変数 HOME が指すディレクトリの下の jageocoder/db
- どちらの環境変数も指定されていない場合は None

戻り値

ディレクトリの絶対パス、または None。

戻り値の型

str

pygeonlp.api.**get_version**() → str

Return version string.

pygeonlp.api.**init**(*db_dir=None*, ***options*)

API をデフォルトパラメータで初期化します。

パラメータ

db_dir (*PathLike, optional*) -- データベースディレクトリ。省略した場合は api.get_db_dir() が返す値を利用します。

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
```

注釈: 実際には、この関数は初期化した Workflow オブジェクト _default_workflow および DictManager オブジェクト _default_manager を作成し、利用可能な状態にします。pygeonlp.api の各関数は、これらのオブジェクトのメンバ関数を呼びだすヘルパー関数として実装されています。

Workflow のパラメータを変更したい場合、 api.default_workflow() で Workflow オブジェクトを取得して set_parser() などを呼びだします。

異なるパラメータで初期化した Workflow オブジェクトを生成し、直接そのメンバ関数を呼びだすことも可能です。

---

pygeonlp.api.**ma_parse**(*sentence*)

センテンスを形態素解析した結果を MeCab 互換の文字列として返します。*deprecated* この関数は低水準 API なので、廃止予定です。

パラメータ

sentence (*str*) -- 解析する文字列。

戻り値

解析結果の改行区切りテキスト。

戻り値の型

str

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> print(api.ma_parse('今日は国会議事堂前まで歩きました。'))

今日　　名詞,副詞可能,*,*,*,*,今日,キョウ,キョー
は　　助詞,係助詞,*,*,*,*,は,ハ,ワ
国会議事堂前　　名詞,固有名詞,地名語,Bn4q6d:国会議事堂前駅/cE8W4w:国会議事堂前駅,*,*,国会議事堂前,,
まで　　助詞,副助詞,*,*,*,*,まで,マデ,マデ
歩き　　動詞,自立,*,*,五段・カ行イ音便,連用形,歩く,アルキ,アルキ
まし　　助動詞,*,*,*,特殊・マス,連用形,ます,マシ,マシ
た　　助動詞,*,*,*,特殊・タ,基本形,た,タ,タ
。　　記号,句点,*,*,*,*,。,。,。
EOS
```

pygeonlp.api.**ma_parseNode**(*sentence*)

センテンスを形態素解析した結果を MeCab 互換のノード配列として返します。*deprecated* この関数は低水準 API なので、廃止予定です。

パラメータ

sentence (*str*) -- 解析する文字列。

戻り値

解析結果のリスト。

戻り値の型

list

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.ma_parseNode('今日は国会議事堂前まで歩きました。')
[{'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '*', 'pos':
↪'BOS/EOS', 'prononciation': '*', 'subclass1': '*', 'subclass2': '*', 'subclass3':
↪'*', 'surface': '', 'yomi': '*'}, {'conjugated_form': '*', 'conjugation_type': '*
↪', 'original_form': '今日', 'pos': '名詞', 'prononciation': 'キョー', 'subclass1
↪': '副詞可能', 'subclass2': '*', 'subclass3': '*', 'surface': '今日', 'yomi': 'キョ
ウ'}, {'conjugated_form': '*', 'conjugation_type': '*', 'original_form': 'は', 'pos
```

(次のページに続く)

(前のページからの続き)

```
↪': '助詞', 'prononciation': 'ワ', 'subclass1': '係助詞', 'subclass2': '*',
↪'subclass3': '*', 'surface': 'は', 'yomi': 'ハ'}, {'conjugated_form': '*',
↪'conjugation_type': '*', 'original_form': '国会議事堂前', 'pos': '名詞',
↪'prononciation': '', 'subclass1': '固有名詞', 'subclass2': '地名語', 'subclass3':
↪'Bn4q6d:国会議事堂前駅/cE8W4w:国会議事堂前駅', 'surface': '国会議事堂前', 'yomi': ''}
↪, {'conjugated_form': '*', 'conjugation_type': '*', 'original_form': 'まで', 'pos
↪': '助詞', 'prononciation': 'マデ', 'subclass1': '副助詞', 'subclass2': '*',
↪'subclass3': '*', 'surface': 'まで', 'yomi': 'マデ'}, {'conjugated_form': '五段・カ
行イ音便', 'conjugation_type': '連用形', 'original_form': '歩く', 'pos': '動詞',
 'prononciation': 'アルキ', 'subclass1': '自立', 'subclass2': '*', 'subclass3': '*',
 'surface': '歩き', 'yomi': 'アルキ'}, {'conjugated_form': '特殊・マス',
↪'conjugation_type': '連用形', 'original_form': 'ます', 'pos': '助動詞',
↪'prononciation': 'マシ', 'subclass1': '*', 'subclass2': '*', 'subclass3': '*',
↪'surface': 'まし', 'yomi': 'マシ'}, {'conjugated_form': '特殊・タ', 'conjugation_
↪type': '基本形', 'original_form': 'た', 'pos': '助動詞', 'prononciation': 'タ',
↪'subclass1': '*', 'subclass2': '*', 'subclass3': '*', 'surface': 'た', 'yomi': 'タ
'}, {'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '。', 'pos':
↪'記号', 'prononciation': '。', 'subclass1': '句点', 'subclass2': '*', 'subclass3
↪': '*', 'surface': '。', 'yomi': '。'}, {'conjugated_form': '*', 'conjugation_type
↪': '*', 'original_form': '*', 'pos': 'BOS/EOS', 'prononciation': '*', 'subclass1
↪': '*', 'subclass2': '*', 'subclass3': '*', 'surface': '', 'yomi': '*'}]
```

pygeonlp.api.**removeDictionary**(*identifier*)

identifier で指定した辞書をデータベースから削除します。

パラメータ

identifier (*str*) -- 辞書の identifier ("geonlp:xxxxx")。

戻り値

常に True。

戻り値の型

bool

例外

RuntimeError -- 指定した辞書が登録されていない場合は RuntimeError が発生します。

pygeonlp.api.**saveDictionaryFromWeb**(*jsonfile*, *csvfile*, *url*, *params=None*, ***kwargs*)

指定した URL にあるページに含まれる辞書メタデータ（JSON-LD）を取得し、メタデータに記載されている URL から地名解析辞書（CSV ファイル）を取得し、指定されたファイルに保存します。

パラメータ

- **jsonfile** (*str*) -- json-ld を保存するファイル名。
- **csvfile** (*str*) -- CSV データを保存するファイル名。
- **url** (*str*) -- 辞書メタデータを含むウェブページの URL。
- **params** (*dict, optional*) -- requests.get に渡す params パラメータ。
- ****kwargs** (*dict, optional*) -- requests.get に渡す kwargs パラメータ。

戻り値

常に True。ダウンロードに失敗した場合は例外が発生します。

戻り値の型

bool

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.saveDictionaryFromWeb('geoshape.json', 'geoshape.csv', 'https://geonlp.ex.
↪nii.ac.jp/dictionary/geoshape-city/')
True
>>> import os
>>> os.remove('geoshape.json')
>>> os.remove('geoshape.csv')
```

pygeonlp.api.**searchWord**(*key*)

指定した表記または読みを持つ語の情報を返します。一致する語が辞書に存在しない場合は None を返します。*deprecated* この関数は低水準 API なので、廃止予定です。

パラメータ

key (*str*) -- 語の表記または読み。

戻り値

geolod_id をキー、語の情報を値に持つ dict。

戻り値の型

dict

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.searchWord('国会議事堂前')
{'Bn4q6d': {'body': '国会議事堂前', 'dictionary_id': 3, 'entry_id': 'LrGGxY',
 'geolod_id': 'Bn4q6d', 'hypernym': ['東京地下鉄', '4 号線丸ノ内線'], 'institution_type
↪': '民営鉄道', 'latitude': '35.674845', 'longitude': '139.74534166666666', 'ne_
↪class': '鉄道施設/鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅', ''],
 'dictionary_identifier': 'geonlp:ksj-station-N02'}, 'cE8W4w': {'body': '国会議事堂前
', 'dictionary_id': 3, 'entry_id': '4NFELa', 'geolod_id': 'cE8W4w', 'hypernym': ['東
京地下鉄', '9 号線千代田線'], 'institution_type': '民営鉄道', 'latitude': '35.
↪67354333333335', 'longitude': '139.74305333333334', 'ne_class': '鉄道施設/鉄道駅',
 'railway_class': '普通鉄道', 'suffix': ['駅', ''], 'dictionary_identifier':
↪'geonlp:ksj-station-N02'}}
```

pygeonlp.api.**setActiveClasses**(*patterns=None*)

解析対象とする固有名クラスの正規表現リストを指定します。いずれかの正規表現に一致する固有名クラスは解析対象となります。

'-' から始まる場合、その正規表現に一致する固有名クラスは対象外となります。

パラメータ

patterns (*list, optional*) -- 解析対象とする固有名クラス（str）のリスト。省略した場合 ['.*'] （全固有名クラス）を対象とします。

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.getActiveClasses()
['.*']
>>> api.searchWord('東京都')
{'QknGsa': {...'dictionary_identifier': 'geonlp:geoshape-pref'}}
>>> api.setActiveClasses(['.*', '-都道府県'])
>>> api.searchWord('東京都')
{}
>>> api.setActiveClasses()
>>> api.getActiveClasses()
['.*']
```

pygeonlp.api.**setActiveDictionaries**(*idlist=None*, *pattern=None*)

インストール済み辞書のうち、解析に利用する辞書を指定します。

パラメータ

- **idlist** (*list, optional*) -- 利用する辞書の id または identifier のリスト。
- **pattern** (*str, optional*) -- 利用する辞書の identifier の正規表現。

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.setActiveDictionaries(pattern=r'geonlp:geoshape')
>>> sorted([x.get_identifier() for x in api.getActiveDictionaries()])
['geonlp:geoshape-city', 'geonlp:geoshape-pref']
```

注釈: idlist と pattern のどちらかは指定する必要があります。

---

pygeonlp.api.**setup_basic_database**(*db_dir=None*, *src_dir=None*)

基本的な地名解析辞書を登録したデータベースを作成します。

パラメータ

- **db_dir** (*PathLike, optional*) -- データベースディレクトリを指定します。ここにデータベースファイルを作成します。既にデータベースが存在する場合は追記します（つまり、既に登録済みの地名解析辞書のデータは失われません）。

  省略された場合には get_db_dir() で決定します。

- **src_dir** (*PathLike, optional*) -- 地名解析辞書ファイルが配置されているディレクトリを指定します。省略された場合には、 sys.prefix または site.USER_BASE の下に pygeonlp_basedata がないか探します。見つからない場合は RuntimeError を送出しますので、ディレクトリを指定してください。

pygeonlp.api.**updateIndex**()

辞書のインデックスを更新して検索可能にします。

14.2 サブモジュール

14.2.1 pygeonlp.api.dict_manager module

class pygeonlp.api.dict_manager.**DictManager**(*db_dir: str | bytes | PathLike | None = None*)

ベースクラス: object

地名語辞書の管理を行なうクラスです。

_dict_cache

辞書 ID（整数）をキー、辞書 identifier（文字列）を値にもつマッピングテーブル（キャッシュ用）。

Type
dict

capi_ma

C 実装の拡張形態素解析機能を利用するためのオブジェクト。

Type
pygeonlp.capi オブジェクト

db_dir

管理対象データベースディレクトリのパス。

Type
str

addDictionaryFromCsv(*csvfile*, *name=None*, *identifier=None*)

指定したパスにある地名解析辞書（CSV ファイル）をデータベースに登録します。

辞書の name と identifier を省略した場合、CSV ファイル名から自動的に設定されます。

既に同じ identifier を持つ辞書データがデータベースに登録されている場合、削除してから新しい辞書データを登録します。

パラメータ

- **csvfile** (*str*) -- 地名解析辞書ファイルのパス。
- **name** (*str, optional*) -- 辞書名。省略した場合、 CSV ファイルの basename を利用します。
- **identifier** (*str, optional*) -- 辞書 identifier。省略した場合、CSV ファイルの basename を取り、geonlp:<basename> を利用します。

戻り値

常に True。登録に失敗した場合は例外が発生します。

戻り値の型

bool

サンプル

```
>>> from pygeonlp.api.dict_manager import DictManager
>>> manager = DictManager()
>>> manager.addDictionaryFromCsv(
...   'base_data/ksj-station-N02-2020.csv',
...   name='日本の鉄道駅（2020年）')
True
>>> manager.updateIndex()
True
```

addDictionaryFromFile(*jsonfile*, *csvfile*)

指定したパスにある辞書メタデータ（JSON ファイル）と地名解析辞書（CSV ファイル）をデータベースに登録します。

既に同じ identifier を持つ辞書データがデータベースに登録されている場合、削除してから新しい辞書データを登録します。

パラメータ

- **jsonfile** (*str*) -- 辞書メタデータファイルのパス。
- **csvfile** (*str*) -- 地名解析辞書ファイルのパス。

戻り値

常に True。登録に失敗した場合は例外が発生します。

戻り値の型

bool

サンプル

```
>>> from pygeonlp.api.dict_manager import DictManager
>>> manager = DictManager()
>>> manager.addDictionaryFromFile(
...   'base_data/geoshape-city.json', 'base_data/geoshape-city.csv')
True
>>> manager.updateIndex()
True
```

addDictionaryFromWeb(*url*, *params=None*, ***kwargs*)

指定した URL にあるページに含まれる辞書メタデータ（JSON-LD）を取得し、メタデータに記載されている URL から地名解析辞書（CSV ファイル）を取得し、データベースに登録します。

既に同じ identifier を持つ辞書データがデータベースに登録されている場合、削除してから新しい辞書データを登録します。

パラメータ

- **url** (*str*) -- 辞書メタデータを含むウェブページの URL。
- **params** (*dict, optional*) -- requests.get に渡す params パラメータ。
- ****kwargs** (*dict, optional*) -- requests.get に渡す kwargs パラメータ。

戻り値

常に True。登録に失敗した場合は例外が発生します。

戻り値の型

bool

サンプル

```
>>> from pygeonlp.api.dict_manager import DictManager
>>> manager = DictManager()
>>> manager.clearDatabase()
True
>>> manager.addDictionaryFromWeb('https://geonlp.ex.nii.ac.jp/dictionary/
↪geoshape-city/')
True
>>> manager.updateIndex()
True
```

clearDatabase()

データベースに登録されている辞書をクリアします。データベース内の地名語も全て削除されます。

この関数は、データベースを作り直す際に利用します。

サンプル

```
>>> from pygeonlp.api.dict_manager import DictManager
>>> manager = DictManager()
>>> manager.clearDatabase()
True
>>> manager.addDictionaryFromWeb(
...   'https://geonlp.ex.nii.ac.jp/dictionary/geoshape-city/')
True
>>> manager.updateIndex()
True
```

getDictionaries()

インストール済み辞書のメタデータ一覧を返します。

戻り値

Metadata インスタンスのリスト。

戻り値の型

list

サンプル

```
>>> from pygeonlp.api.dict_manager import DictManager
>>> manager = Manager()
>>> sorted([x.get_identifier() for x in manager.getDictionaries()])
['geonlp:geoshape-city', 'geonlp:geoshape-pref', 'geonlp:ksj-station-N02']
```

getDictionary(*id_or_identifier*)

指定した id または identifier を持つ辞書のメタデータを返します。id, identifier に一致する辞書が存在しない場合は None を返します。

パラメータ

id_or_identifier (*str or int*) -- str の場合は辞書 identifier で指定。int の場合は内部辞書 id で指定。

戻り値

Metadata インスタンス。

戻り値の型

Metadata

サンプル

```
>>> from pygeonlp.api.dict_manager import DictManager
>>> manager = DictManager()
>>> manager.getDictionary('geonlp:ksj-station-N02')
{"@context": "https://schema.org/", "@type": "Dataset", "alternateName": "",
↪"creator": [{"@type": "Organization", "name": "株式会社情報試作室", "sameAs":
↪"https://www.info-proto.com/"}], "dateModified": "2021-08-27T17:18:18+09:00",
 "description": "国土数値情報「鉄道データ（N02)」から作成した、日本の鉄道駅（地下鉄を含む）
の辞書です。hypernym には運営者名と路線名を記載しています。「都営」ではなく「東京都」のよう
になっていますので注意してください。自由フィールドとして、railway_class に「鉄道区分」、
institution_type に「事業者種別」を含みます。", "distribution": [{"@type":
↪"DataDownload", "contentUrl": "https://www.info-proto.com/static/ksj-station-
↪N02.csv", "encodingFormat": "text/csv"}], "identifier": ["geonlp:ksj-station-
↪N02"], "isBasedOn": {"@type": "CreativeWork", "name": "国土数値情報 鉄道データ",
↪"url": "https://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-N02-v2_2.html"},
↪"keywords": ["GeoNLP", "地名辞書"], "license": "https://creativecommons.org/
↪licenses/by/4.0/", "name": "国土数値情報 鉄道データ（駅）", "size": "10191",
↪"spatialCoverage": {"@type": "Place", "geo": {"@type": "GeoShape", "box": "26.
↪193265 127.652285 45.41616333333333 145.59723"}}, "temporalCoverage": "../..",
↪"url": "https://www.info-proto.com/static/ksj-station-N02.html"}
```

static get_package_files()

pip list コマンドを実行し、pygeonlp パッケージとしてインストールされたファイルのフルパス一覧を取得します。

参考： https://pip.pypa.io/en/latest/user_guide/#using-pip-from-your-program

init_capi_ma()

removeDictionary(*identifier*)

identifier で指定した辞書をデータベースから削除します。

パラメータ

identifier (*str*) -- 辞書の identifier ("geonlp:xxxxx")。

戻り値

常に True。

戻り値の型

bool

例外

RuntimeError -- 指定した辞書が登録されていない場合は RUntimeError が発生します。

サンプル

```
>>> from pygeonlp.api.dict_manager import DictManager
>>> manager = DictManager()
>>> manager.removeDictionary('geonlp:geoshape-city')
True
>>> manager.updateIndex()
True
```

saveDictionaryFromWeb(*jsonfile*, *csvfile*, *url*, *params=None*, ***kwargs*)

指定した URL にあるページに含まれる辞書メタデータ（JSON-LD）を取得し、メタデータに記載されている URL から地名解析辞書（CSV ファイル）を取得し、指定されたファイルに保存します。

パラメータ

- **jsonfile** (*str*) -- json-ld を保存するファイル名。
- **csvfile** (*str*) -- CSV データを保存するファイル名。
- **url** (*str*) -- 辞書メタデータを含むウェブページの URL。
- **params** (*dict, optional*) -- requests.get に渡す params パラメータ。
- ****kwargs** (*dict, optional*) -- requests.get に渡す kwargs パラメータ。

戻り値

常に True。ダウンロードに失敗した場合は例外が発生します。

戻り値の型

bool

サンプル

```
>>> from pygeonlp.api.dict_manager import DictManager
>>> manager = DictManager()
>>> manager.saveDictionaryFromWeb('geoshape.json', 'geoshape.csv', 'https://
↪geonlp.ex.nii.ac.jp/dictionary/geoshape-city/')
True
>>> import os
>>> os.remove('geoshape.json')
>>> os.remove('geoshape.csv')
```

setupBasicDatabase(*src_dir=None*)

基本的な地名解析辞書を登録したデータベースを作成します。

パラメータ

src_dir (*str, optional*) -- 地名解析辞書ファイルが配置されているディレクトリ。省略された場合には、 sys.prefix または site.USER_BASE の下に pygeonlp_basedata がないか探します。見つからない場合は RuntimeError を送出しますので、ディレクトリを指定してください。

updateIndex()

辞書のインデックスを更新して検索可能にします。

サンプル

```
>>> from pygeonlp.api.dict_manager import DictManager
>>> manager = DictManager()
>>> manager.clearDatabase()
True
>>> manager.addDictionaryFromWeb('https://geonlp.ex.nii.ac.jp/dictionary/
↪geoshape-city/')
True
>>> manager.updateIndex()
True
```

14.2.2 pygeonlp.api.dictionary module

class pygeonlp.api.dictionary.**Dictionary**(*metadata*, *csvtext*)

ベースクラス: object

GeoNLP 辞書の管理クラス辞書メタデータと CSV データに対する操作を行ないます。

通常、開発者がこのクラスを直接操作する必要はありません。

metadata

メタデータのインスタンス

Type

Metadata

csvtext

CSV テキスト

Type

str

__init__(*metadata*, *csvtext*)

与えられたパラメータでクラス変数を初期化します。

パラメータ

- **metadata** (Metadata) -- メタデータのインスタンス
- **csvtext** (*str*) -- CSV テキスト

add(*capi_ma*)

システムに辞書を登録します。

パラメータ

capi_ma (capi.MA) -- 辞書を管理する MA オブジェクト。init() で初期化済みである必要があります。

戻り値

常に True が返ります。失敗した場合は例外が発生します。

戻り値の型

bool

classmethod create(*csvfile*, *name=None*, *identifier=None*)

指定したパスにある地名解析辞書（CSV ファイル）を読み込み Dictionary インスタンスを作成します。

パラメータ

- **csvfile** (*str*) -- 地名解析辞書ファイルのパス
- **name** (*str, optional*) -- 辞書名。省略した場合、 CSV ファイルの basename を利用します。
- **identifier** (*str, optional*) -- 辞書 identifier。省略した場合、CSV ファイルの basename を取り、geonlp:<basename> を利用します。

戻り値

作成した Dictionary インスタンス。

戻り値の型

Dictionary

classmethod download(*url*, *params=None*, ***kwargs*)

指定された URL からウェブページをダウンロードし、ヘッダに記載されている json-ld を辞書メタデータとして抽出します。また、メタデータに書かれているデータダウンロード URL にアクセスして CSV データを取得します。取得したメタデータと CSV データを保持する Dictionary インスタンスを作成します。

パラメータ

- **url** (*str*) -- json-ld を含むウェブページの URL
- **params** (*dict, optional*) -- requests.get に渡す params パラメータ
- ****kwargs** -- requests.get に渡す kwargs パラメータ

戻り値

作成した Dictionary インスタンス。

戻り値の型

Dictionary

get_identifier()

この辞書の identifier を返します。

戻り値

辞書 identifier 文字列。

戻り値の型

str

get_name()

この辞書の名前を返します。

戻り値

辞書の名前。

戻り値の型

str

classmethod load(*jsonfile*, *csvfile*)

指定したパスにある辞書メタデータ（JSON ファイル）と地名解析辞書（CSV ファイル）を読み込み Dictionary インスタンスを作成します。

パラメータ

- **jsonfile** (*str*) -- 辞書メタデータファイルのパス
- **csvfile** (*str*) -- 地名解析辞書ファイルのパス

戻り値

作成した Dictionary インスタンス。

戻り値の型

Dictionary

save(*jsonfile*, *csvfile*)

この辞書が保持している json-ld と CSV データをファイルに保存します。

パラメータ

- **jsonfile** (*str*) -- json-ld を保存するファイル名
- **csvfile** (*str*) -- CSV データを保存するファイル名

戻り値

常に True が返ります。失敗した場合は例外が発生します。

戻り値の型

bool

class pygeonlp.api.dictionary.**DictionaryError**

ベースクラス: RuntimeError

辞書データの操作の際に例外が起こると、このクラスが発生します。

14.2.3 pygeonlp.api.filter module

class pygeonlp.api.filter.**Filter**(**kwargs*)

ベースクラス: object

ラティス表現を入力として、条件に一致しないノードを除外した結果をラティス表現として出力するフィルタ機能の基底クラス。

このクラスは全てのフィルタに共通の機能を提供するものなので、派生クラスを利用してください。

__init__(**kwargs*)

フィルタを初期化します。初期化の内容は派生するクラスごとに拡張する必要があります。

when_all_failed

全ての候補が条件を満たさなかった場合の挙動を指定します。

"return_all"

全ての候補をそのまま残します。

"convert_to_normal"

'名詞・固有名詞・地域・一般' ノードを 1 つ作成して返します。

Type

str

apply(*input*, **kwargs*)

入力データの各形態素に対して apply_filter() を適用し、合格したノードだけを残した結果を返します。

apply_filter() では一つの形態素に対応する候補ノードのリストに含まれる情報のみを利用してフィルタ処理を行なうため、前後の形態素を見て判定する必要があるフィルタクラスでは、apply() メソッドを上書きする必要があります。

パラメータ

input (*list*) -- 入力となるラティス表現。

戻り値

filter_func() を適用して要素を削除されたラティス表現。

戻り値の型

list

apply_filter(*candidates*, *default=None*, ***kwargs*)

ラティス表現の 1 つの形態素に対する複数の候補ノードに対して順番に filter_func() を適用し、条件に一致しないノードを削除した候補ノードのリストを返します。

default で指定された地名語 ID を持つノードが条件に一致した場合、結果候補ノードリストの先頭に配置します。

filter_func() では個々の候補ノードに含まれる情報のみを利用してフィルタへの合否判定処理を行なうため、1 つの形態素に対する候補ノードを比較して最適なものを選択する必要があるフィルタクラスでは、apply_filter() メソッドを上書きする必要があります。

パラメータ

- **candidates** (*list*) -- 候補ノードのリスト
- **default** (*str, optional*) -- 最優先候補としたいノードの geolod_id
- ****kwargs** -- キーワードパラメータ、派生クラスでは利用する可能性があります。

戻り値

条件に一致した候補ノードのリスト。

戻り値の型

list

static count_geowords(*lattice*)

ラティス表現を高速に粗く解析し、地名語候補を含む形態素の数や ne_class の分布などを集計します。

詳細な解析を行うためのプランを決定する前処理として利用します。

パラメータ

lattice (*list*) -- ラティス表現。

戻り値

以下の要素を含む集計結果を返します。

num_geowords: int 地
名語候補を 1 つ以上含む形態素ノードの数。

num_addresses: int 住
所候補を 1 つ以上含む形態素ノードの数。

ne_classes: dict 固
有名クラスをキー、そのクラスの地名語を 1 つ以上含む形態素ノードの数を値とする dict。

戻り値の型
dict

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> parser = api.parser.Parser()
>>> api.filter.Filter.count_geowords(
...   parser.analyze_sentence('国会議事堂前まで歩きました。'))
{'num_geowords': 1, 'num_addresses': 0,
   'ne_classes': {'鉄道施設/鉄道駅': 1, '鉄道施設': 1}}
```

filter_func()

候補ノードが条件に一致するかどうかを判定します。派生クラスでそれぞれ実装する必要があります。

パラメータ
candidate (Node) -- 判定する形態素ノード。

戻り値 基
底クラスでは常に True を返します。

戻り値の型
bool

class pygeonlp.api.filter.**EntityClassFilter**(*ne_class*, ***kwargs*)

ベースクラス: *Filter*

形態素ごとの候補ノード集合の中に、指定した固有名クラスと一致する地名語が含まれていれば、それ以外の候補は削除するフィルタ。テキストに含まれている地名語の固有名クラスが事前に分かっていて、そのクラスに一致する地名語だけを抽出したい場合などに利用します。

候補ノード集合の中に一致する地名語が無い場合は候補リストをそのまま返します。

__init__(*ne_class*, ***kwargs*)

パラメータ

ne_class (*str*) -- このフィルタで選択したい固有名クラスを表す正規表現文字列。例： r'国/?.*'

filter_func(*candidate*, ***kwargs*)

パラメータ

candidate (Node) -- 対象の形態素ノード

戻り値

ne_class に一致する場合は True, 一致しない場合は Falase

戻り値の型

bool

class pygeonlp.api.filter.**GreedySearchFilter**(*scoring_class=None*, *scoring_options=None*, ***kwargs*)

ベースクラス: *Filter*

前後の地名語との関係を見て最もスコアが高くなる地名語を 1 つだけ残し、それ以外は削除するフィルタ。

ただし同点となる地名語が複数存在する場合は複数の候補を残します。

__init__(*scoring_class=None*, *scoring_options=None*, ***kwargs*)

パラメータ

- **scoring_class** (*class, optional*) -- パスのスコアとノード間のスコアを計算する関数を持つスコアリングクラス。指定しない場合、pygeonlp.api.scoring モジュール内の ScoringClass が利用されます。
- **scoring_options** (*any, optional*) -- スコアリングクラスの初期化に渡すオプションパラメータ。

apply(*input*, ***kwargs*)

このフィルタは前後の形態素ノードを調べて地名語間の関係を計算するため、apply() をオーバーライドします。

パラメータ

input (*list*) -- 入力となるラティス表現。

戻り値

フィルタを適用したラティス表現。

戻り値の型

list

class pygeonlp.api.filter.**InputBasedFilter**(**kwargs)

ベースクラス: *Filter*

入力によって対象とする地名語クラスやパス表現の選択ロジックを変更して、実時間で処理可能な候補に絞り込むフィルタ。

__init__(**kwargs)

フィルタを初期化します。このクラスには指定するパラメータはありません。

apply(*lattice*, **kwargs)

入力ベースフィルタを適用します。

パラメータ

lattice (*list*) -- 入力となるラティス表現。

戻り値

フィルタ適用後のラティス表現。

戻り値の型

list

注釈: 時刻表や地震速報など、特定の地名語クラスを持つ地名が高い密度で出現する文字列を解析する場合、候補となるクラスを絞り込んで解析処理時間を短縮し、精度を向上させます。

---

class pygeonlp.api.filter.**FilterError**

ベースクラス: RuntimeError

フィルタ処理の際に例外が起こると、このクラスが発生します。

14.2.4 pygeonlp.api.linker module

class pygeonlp.api.linker.**LinkedResults**(*lattice*)

ベースクラス: object

ラティス表現からパス表現の候補を次々に出力するイテレータを作成します。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.linker import LinkedResults
>>> api.init()
>>> for lr in LinkedResults(api.analyze('国会議事堂前まで歩きました。')):
...     [x.simple() for x in lr]
["国会議事堂前(GEOWORD:['東京地下鉄', '4号線丸ノ内線'])", 'まで(NORMAL)', '歩き
(NORMAL)', 'まし(NORMAL)', 'た(NORMAL)', '。(NORMAL)']
["国会議事堂前(GEOWORD:['東京地下鉄', '9号線千代田線'])", 'まで(NORMAL)', '歩き
(NORMAL)', 'まし(NORMAL)', 'た(NORMAL)', '。(NORMAL)']
```

counter()

カウンターを返します。

get_result()

カウンターが示している候補リストを返します。カウンターは次に進めません。

increment_counter()

カウンターを次に進めます。

reset_counter()

カウンターを初期化します。

class pygeonlp.api.linker.**Evaluator**(*scoring_class=None*, *scoring_options=None*, *max_results=5*, *max_combinations=None*)

ベースクラス: object

ラティス表現から、指定したメソッドで計算したスコアの高いものから順に並べたパス表現を作成します。

スコアリングに独自のメソッドを利用する場合は scoring_method で指定してください。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.linker import Evaluator
>>> api.init()
>>> rr = Evaluator(max_results=5)
>>> for x in rr.get(api.analyze('福島は大阪から2分です。')):
...   (x['score'], [n.simple() for n in x['result']])
...
(46, ["福島(GEOWORD:['西日本旅客鉄道', '大阪環状線'])", 'は(NORMAL)', "大阪(GEOWORD:[
```

(次のページに続く)

(前のページからの続き)

```
↪' 西日本旅客鉄道', ' 東海道線'])", ' から (NORMAL)', '2(NORMAL)', ' 分 (NORMAL)', ' です
(NORMAL)', '。(NORMAL)'])
(46, ["福島 (GEOWORD:[' 西日本旅客鉄道', ' 大阪環状線'])", ' は (NORMAL)', "大阪 (GEOWORD:[
↪' 西日本旅客鉄道', ' 大阪環状線'])", ' から (NORMAL)', '2(NORMAL)', ' 分 (NORMAL)', ' です
(NORMAL)', '。(NORMAL)'])
(41, ["福島 (GEOWORD:[' 阪神電気鉄道', ' 本線'])", ' は (NORMAL)', "大阪 (GEOWORD:[' 西日本
旅客鉄道', ' 東海道線'])", ' から (NORMAL)', '2(NORMAL)', ' 分 (NORMAL)', ' です (NORMAL)',
 '。(NORMAL)'])
(41, ["福島 (GEOWORD:[' 阪神電気鉄道', ' 本線'])", ' は (NORMAL)', "大阪 (GEOWORD:[' 西日本
旅客鉄道', ' 大阪環状線'])", ' から (NORMAL)', '2(NORMAL)', ' 分 (NORMAL)', ' です (NORMAL)
↪', '。(NORMAL)'])
(36, ["福島 (GEOWORD:[' 福島交通', ' 飯坂線'])", ' は (NORMAL)', "大阪 (GEOWORD:[' 西日本旅
客鉄道', ' 東海道線'])", ' から (NORMAL)', '2(NORMAL)', ' 分 (NORMAL)', ' です (NORMAL)',
↪'。(NORMAL)'])
```

scoring_class

スコアリングを行なうクラス。

Type

class instance

scoring_options

スコアリングクラスの初期化に渡すオプションパラメータ。

Type

any

scorer

スコアリングを行なうクラスインスタンス。

Type

service.ScoringClass instance

max_results

保持する結果の最大数。

Type

int

__init__(*scoring_class=None*, *scoring_options=None*, *max_results=5*, *max_combinations=None*)

パラメータ

- **scoring_class** (*class*, *optional*) -- パスのスコアとノード間のスコアを計算する関数を持つスコアリングクラス。指定しない場合、pygeonlp.api.scoring モジュール内の ScoringClass が利用されます。
- **scoring_options** (*any*, *optional*) -- スコアリングクラスの初期化に渡すオプションパラメータ。
- **max_results** (*int*, *optional*) -- 保持する結果の最大数を指定します（デフォルト = 5）。
- **max_combinations** (*int*, *optional*) -- ノード候補の組み合わせ数の上限値。これを超える組み合わせが可能な入力が与えられた場合は例外 LinkerError を発生します。デフォルト値は linker.MAX_COMBINATIONS です。

as_dict(*lattice*)

get() と同じ処理を行ないますが、結果に含まれるノードの情報を JSON に変換可能な dict に変換してから返します。

パラメータ

lattice (*list*) -- 入力となるラティス表現。

戻り値

get() の出力結果を JSON 変換可能な形式に変換したリスト。

戻り値の型

list

as_geojson(*lattice*)

get() と同じ処理を行ないますが、結果に含まれるノードの情報を GeoJSON FeatureCollection に変換可能な dict に変換してから返します。

パラメータ

lattice (*list*) -- 入力となるラティス表現。

戻り値

get() の出力結果を GeoJSON 変換可能な形式に変換したリスト。

戻り値の型

list

static collect_addresses(*result*)

パス表現の結果に含まれる住所セットを返します。

処理結果に含まれる住所だけを列挙する場合に利用する簡易メソッドです。

static collect_geowords(*result*)

パス表現の結果に含まれる地名語のセットを返します。

処理結果に含まれる地名語だけを列挙する場合に利用する簡易メソッドです。

count_combinations(*lattice*)

ラティス形式の入力に対し、組み合わせたパス表現の個数を計算します。

パラメータ

lattice (*list*) -- 入力となるラティス表現。

戻り値

組み合わせの数。

戻り値の型

int

get(*lattice*)

ラティス表現を入力として、スコアリングと並べ替えを行ないます。

パラメータ

lattice (*list*) -- 入力となるラティス表現。

戻り値

スコアを 'score', パス表現の解析結果を 'result' に持つ dict のリスト。スコア降順にソートされ、最大 max_results 個の要素を含みます。

戻り値の型

list

14.2.5 pygeonlp.api.metadata module

class pygeonlp.api.metadata.**Metadata**(*jsonld=None*)

ベースクラス: object

JSON-LD 形式メタデータの管理クラス。

jsonld

メタデータ json-ld 文字列。

Type

str

__init__(*jsonld=None*)

パラメータ

jsonld (*str or dict*) -- json-ld 文字列、またはデコードした dict。

classmethod download(*url*, *params=None*, ***kwargs*)

指定した URL からウェブページをダウンロードし、ヘッダに記載されている json-ld を辞書メタデータとして抽出します。

パラメータ

- **url** (*str*) -- json-ld を含むウェブページの URL。
- **params** (*dict*, *optional*) -- requests.get に渡す params パラメータ。
- ****kwargs** (**キーワードパラメータ**, *optional*) -- requests.get に渡す kwargs パラメータ。

戻り値

抽出した json-ld 文字列をセットした Metadata インスタンス

戻り値の型

Metadata

例外

MetadataError -- ウェブページのダウンロードに失敗した場合。ウェブページから json-ld が見つけられない場合。ウェブページに複数の json-ld が含まれている場合。

download_csv(*content_url=None*, *params=None*, ***kwargs*)

content_url から CSV をダウンロードします。

パラメータ

- **content_url** (*str*, *optional*) -- CSV をダウンロードする URL を指定します。省略された場合、 json-ld から content_url を抽出します。
- **params** (*dict*, *optional*) -- requests.get に渡す params パラメータ。
- ****kwargs** (**キーワードパラメータ**, *optional*) -- requests.get に渡す kwargs パラメータ。

戻り値

ダウンロードした CSV テキスト。

戻り値の型

str

例外

MetadataError -- CSV のダウンロードに失敗した場合。

get_content_url()

json-ld 文字列から、コンテンツ CSV をダウンロードする content_url を抽出します。

戻り値

抽出した content_url 文字列。

戻り値の型

str

例外

MetadataError -- json-ld がセットされていない場合。json-ld から content_url が見つけられない場合。

`get_identifier`()

json-ld 文字列から GeoNLP identifier を抽出します。

GeoNLP identifier は 'geonlp:' で始まる必要があります。

戻り値

抽出した identifier 文字列。

戻り値の型

str

例外

MetadataError -- json-ld がセットされていない場合。json-ld から identifier が見つけられない場合。

`get_name`()

json-ld 文字列からデータセットの名前を抽出します。

戻り値

データセットの名前。

戻り値の型

str

例外

MetadataError -- json-ld がセットされていない場合。json-ld から identifier が見つけられない場合。

`classmethod load`(*file*)

指定したファイルに含まれる json-ld 文字列から Metadata インスタンスを作成します。

パラメータ

file (*str*) -- json-ld 文字列を含むファイル名。

戻り値

json-ld 文字列をセットした Metadata インスタンス。

戻り値の型

Metadata

classmethod loads(*jsonld*)

指定した json-ld 文字列から Metadata インスタンスを作成します。

パラメータ

jsonld (*str*) -- json-ld 文字列。

戻り値

json-ld 文字列をセットした Metadata インスタンス。

戻り値の型

Metadata

class pygeonlp.api.metadata.**MetadataError**

ベースクラス: RuntimeError

メタデータの操作の際に例外が起こると、このクラスが発生します。

14.2.6 pygeonlp.api.node module

class pygeonlp.api.node.**Node**(*surface*, *node_type*, *morphemes*, *geometry=None*, *prop=None*)

ベースクラス: object

形態素ノードを表すクラス。

service.analyze() が返すラティス表現や、 linker.Evaluator.get() が返すパス表現では、形態素は Node インスタンスとして格納されます。

surface

表記文字列。

Type

str

node_type

ノードの種類。地名語ノードは 1、住所ノードは 2、それ以外は 0 です。

Type

int

morphemes

形態素の属性。

Type

list of dict

geometry

地理属性。

Type
object or None

prop

その他の属性（geoword または住所）。

Type
dict

ADDRESS = 2

GEOWORD = 1

IGNORE = -1

NORMAL = 0

__init__(*surface*, *node_type*, *morphemes*, *geometry=None*, *prop=None*)

ノード情報を初期化します。

ノードは基本的に Parser が作成したものを利用してください。

as_dict()

ノード情報を JSON 出力可能な dict オブジェクトに変換します。

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.analyze('国会議事堂前')[0][0].as_dict()
{'surface': '国会議事堂前', 'node_type': 'GEOWORD', 'morphemes': {'conjugated_form
↪': '*', 'conjugation_type': '*', 'original_form': '国会議事堂前', 'pos': '名詞',
 'prononciation': '', 'subclass1': '固有名詞', 'subclass2': '地名語', 'subclass3
↪': 'Bn4q6d:国会議事堂前駅', 'surface': '国会議事堂前', 'yomi': ''}, 'geometry': {
↪'type': 'Point', 'coordinates': [139.74534166666666, 35.674845]}, 'prop': {
↪'body': '国会議事堂前', 'dictionary_id': 3, 'entry_id': 'LrGGxY', 'geolod_id':
↪'Bn4q6d', 'hypernym': ['東京地下鉄', '4 号線丸ノ内線'], 'institution_type': '民営
鉄道', 'latitude': '35.674845', 'longitude': '139.74534166666666', 'ne_class': '
鉄道施設/鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅', ''], 'dictionary_
↪identifier': 'geonlp:ksj-station-N02'}}
```

戻り値

JSON 出力可能な dict オブジェクト。

戻り値の型

dict

`as_geojson`()

ノード情報を GeoJSON の Feature 形式にダンプ可能な dict オブジェクトに変換します。

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.analyze('国会議事堂前')[0][0].as_geojson()
{'type': 'Feature', 'geometry': {'type': 'Point', 'coordinates': [139.
↪74534166666666, 35.674845]}, 'properties': {'surface': '国会議事堂前', 'node_
↪type': 'GEOWORD', 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*
↪', 'original_form': '国会議事堂前', 'pos': '名詞', 'prononciation': '',
 'subclass1': '固有名詞', 'subclass2': '地名語', 'subclass3': 'Bn4q6d:国会議事堂前
駅', 'surface': '国会議事堂前', 'yomi': ''}, 'geoword_properties': {'body': '国会
議事堂前', 'dictionary_id': 3, 'entry_id': 'LrGGxY', 'geolod_id': 'Bn4q6d',
↪'hypernym': ['東京地下鉄', '4 号線丸ノ内線'], 'institution_type': '民営鉄道',
↪'latitude': '35.674845', 'longitude': '139.74534166666666', 'ne_class': '鉄道施
設/鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅', ''], 'dictionary_
↪identifier': 'geonlp:ksj-station-N02'}}}
```

戻り値

GeoJSON Feature 形式に変換可能な dict オブジェクト。

戻り値の型

dict

`distance`(*node*)

自身と他のノードの距離 (m) を計算します。このメソッドは geographiclib を利用します。https://pypi.org/project/geographiclib/

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> lattice = api.analyze('国会議事堂前,永田町')
>>> n0 = lattice[0][0]
>>> n1 = lattice[2][0]
>>> round(n0.distance(n1), 6)
676.592019
```

get_lonlat()

ノードの経度、緯度を返します。

戻り値

以下の要素を持つ dict。

- lat
  [float] ノードの代表点の緯度。
- lon
  [float] ノードの代表点の経度。

戻り値の型

dict

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> node = api.analyze('国会議事堂前')[0][0]
>>> node.get_lonlat()
{'lat': 35.674845, 'lon': 139.74534166666666}
```

```
>>> import pygeonlp.api as api
>>> api.init()
>>> node = api.analyze('千代田区一ツ橋2-1-2', jageocoder=True)[0][0]
>>> node.get_lonlat()
{'lat': 35.69..., 'lon': 139.75...}
```

get_notations()

このノードの表記として可能性があるものの候補を取得し、set を返します（重複を除去し、AND を高速に計算するため）。

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> # set は順番を保持しないので sorted で昇順にソート
>>> sorted(api.analyze('国会議事堂前')[0][0].get_notations())
['国会議事堂前', '国会議事堂前駅']
```

注釈: 何度も計算しないように、計算した結果は _attr['notations'] に保持します。

---

get_point_object()

地名語ノードまたは住所ノードの場合、経度緯度から Point オブジェクトを作成します。

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.analyze('国会議事堂前')[0][0].get_point_object().ExportToWkt()
'POINT (139.745341666667 35.674845)'
```

注釈: gdal がインストールされていない場合は None を返します。何度も計算しないように、計算した結果は _attr['point'] に保持します。

---

simple()

ノード情報をシンプルな形式で表現します。

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> api.analyze('国会議事堂前')[0][0].simple()
"国会議事堂前 (GEOWORD:['東京地下鉄', '4 号線丸ノ内線'])"
```

戻り値 シンプルな形式の文字列。

戻り値の型

str

14.2.7 pygeonlp.api.parser module

class pygeonlp.api.parser.**Parser**(*db_dir=None*, *jageocoder=None*, *address_regex=None*, ***options*)

ベースクラス: object

形態素解析と地名語抽出を行なうパーザ。

service

利用する Service インスタンス。

Type

pygeonlp.service.Service

jageocoder_tree

住所ジオコーダー jageocoder の AddressTree インスタンス。利用しない場合は False を指定してください。

Type

jageocoder.address.AddressTree

address_regex

住所要素の先頭になりえる固有名クラスの正規表現（コンパイル済み）。

Type

regex

scoring_class

パスのスコアリングに利用するクラス。

Type

class

__init__(*db_dir=None*, *jageocoder=None*, *address_regex=None*, ***options*)

パーザを初期化します。

パラメータ

- **db_dir** (*PathLike, optional*) -- データベースディレクトリ。省略した場合は api.init.get_db_dir() が返す値を利用します。
- **jageocoder** (*jageocoder.tree.AddressTree, optional*) -- 利用する住所ジオコーダーを指定します。省略した場合、jageocoder モジュールのデフォルトオブジェクトを利用しま

す。False を指定した場合、ジオコーディング機能を利用しません。

- **address_regex** (*str, optional*) -- 住所表記の開始とみなす地名語の固有名クラスを表す正規表現。省略した場合、`r'^(都道府県|市区町村|行政地域|居住地名)(/.+|)'` を利用します。

add_address_candidates(*lattice*, *keep_nodes=False*, ***kwargs*)

ラティス表現に住所候補を追加します。

パラメータ

- **lattice** (*list*) -- analyze_sentence の結果のラティス表現（住所を含まない）。
- **keep_nodes** (*bool, optional*) -- 住所以外のノードを維持するかどうかを指示するフラグ。デフォルトは False （維持しない）。

戻り値

住所候補を追加したラティス表現。

戻り値の型

list

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.devtool import pp_lattice
>>> from pygeonlp.api.node import Node
>>> api.init()
>>> parser = api.parser.Parser(jageocoder=True)
>>> lattice = parser.analyze_sentence('アメリカ大使館：港区赤坂1-10-5')
>>> lattice_address = parser.add_address_candidates(lattice, True)
>>> pp_lattice(lattice_address)
#0:'アメリカ大使館'
  アメリカ大使館(NORMAL)
#1:'：'
  ：(NORMAL)
#2:'港区'
  港区(GEOWORD:['東京都'])
  港区(GEOWORD:['愛知県', '名古屋市'])
  港区(GEOWORD:['大阪府', '大阪市'])
  港区赤坂1-10-(ADDRESS:東京都/港区/赤坂/一丁目/10番)[6]
#3:'赤坂'
  赤坂(GEOWORD:['上毛電気鉄道', '上毛線'])
```

(次のページに続く)

(前のページからの続き)

```
  赤坂 (GEOWORD:['東京地下鉄', '9 号線千代田線'])
  赤坂 (GEOWORD:['富士急行', '大月線'])
  赤坂 (GEOWORD:['福岡市', '1 号線 (空港線)'])
#4:'1'
  1(NORMAL)
#5:'-'
  -(NORMAL)
#6:'10'
  10(NORMAL)
#7:'-'
  -(NORMAL)
#8:'5'
  5(NORMAL)
>>> lattice_address = parser.add_address_candidates(lattice)
>>> pp_lattice(lattice_address)
#0:'アメリカ大使館'
  アメリカ大使館 (NORMAL)
#1:'：'
  ：(NORMAL)
#2:'港区赤坂 1-10-'
  港区赤坂 1-10-(ADDRESS: 東京都/港区/赤坂/一丁目/10 番)[6]
#3:'5'
  5(NORMAL)
>>> node = lattice_address[2][0]
>>> len(node.morphemes)
6
>>> '東京都' in node.morphemes[0].prop['hypernym']
True
>>> node.morphemes[1].node_type == Node.NORMAL
True
```

analyze(*sentence*, ***kwargs*)

文を解析した結果をラティス表現で返します。

パラメータ

sentence (*str*) -- 解析するテキスト。

戻り値

解析結果のラティス表現。

戻り値の型

list

サンプル

```
>>> from pygeonlp.api.parser import Parser
>>> parser = Parser()
>>> parser.analyze('今日は国会議事堂前まで歩きました。')
[[{"surface": "今日", "node_type": "NORMAL", "morphemes": {"conjugated_form": "*
↪", "conjugation_type": "*", "original_form": "今日", "pos": "名詞",
↪"prononciation": "キョー", "subclass1": "副詞可能", "subclass2": "*", "subclass3
↪": "*", "surface": "今日", "yomi": "キョウ"}, "geometry": null, "prop": null}],␣
↪[{"surface": "は", "node_type": "NORMAL", "morphemes": {"conjugated_form": "*",
 "conjugation_type": "*", "original_form": "は", "pos": "助詞", "prononciation":
↪"ワ", "subclass1": "係助詞", "subclass2": "*", "subclass3": "*", "surface": "は
", "yomi": "ハ"}, "geometry": null, "prop": null}], [{"surface": "国会議事堂前",
↪"node_type": "GEOWORD", "morphemes": {"conjugated_form": "*", "conjugation_type
↪": "*", "original_form": "国会議事堂前", "pos": "名詞", "prononciation": "",
↪"subclass1": "固有名詞", "subclass2": "地名語", "subclass3": "Bn4q6d: 国会議事堂前
駅", "surface": "国会議事堂前", "yomi": ""}, "geometry": {"type": "Point",
 "coordinates": [139.74534166666666, 35.674845]}, "prop": {"body": "国会議事堂前",
 "dictionary_id": 3, "entry_id": "LrGGxY", "geolod_id": "Bn4q6d", "hypernym": [
↪"東京地下鉄", "4 号線丸ノ内線"], "institution_type": "民営鉄道", "latitude": "35.
↪674845", "longitude": "139.74534166666666", "ne_class": "鉄道施設/鉄道駅",
↪"railway_class": "普通鉄道", "suffix": ["駅", ""], "dictionary_identifier":
↪"geonlp:ksj-station-N02"}}, {"surface": "国会議事堂前", "node_type": "GEOWORD",
 "morphemes": {"conjugated_form": "*", "conjugation_type": "*", "original_form":
↪"国会議事堂前", "pos": "名詞", "prononciation": "", "subclass1": "固有名詞",
 "subclass2": "地名語", "subclass3": "cE8W4w: 国会議事堂前駅", "surface": "国会議事堂
前", "yomi": ""}, "geometry": {"type": "Point", "coordinates": [139.
↪74305333333334, 35.673543333333335]}, "prop": {"body": "国会議事堂前",
↪"dictionary_id": 3, "entry_id": "4NFELa", "geolod_id": "cE8W4w", "hypernym": [
↪"東京地下鉄", "9 号線千代田線"], "institution_type": "民営鉄道", "latitude": "35.
↪673543333333335", "longitude": "139.74305333333334", "ne_class": "鉄道施設/鉄道
駅", "railway_class": "普通鉄道", "suffix": ["駅", ""], "dictionary_identifier":
↪"geonlp:ksj-station-N02"}}], [{"surface": "まで", "node_type": "NORMAL",
 "morphemes": {"conjugated_form": "*", "conjugation_type": "*", "original_form":
↪"まで", "pos": "助詞", "prononciation": "マデ", "subclass1": "副助詞", "subclass2
↪": "*", "subclass3": "*", "surface": "まで", "yomi": "マデ"}, "geometry": null,
 "prop": null}], [{"surface": "歩き", "node_type": "NORMAL", "morphemes": {
↪"conjugated_form": "五段・カ行イ音便", "conjugation_type": "連用形", "original_
```

(次のページに続く)

(前のページからの続き)

```
↪form": "歩く", "pos": "動詞", "prononciation": "アルキ", "subclass1": "自立",
↪"subclass2": "*", "subclass3": "*", "surface": "歩き", "yomi": "アルキ"},
↪"geometry": null, "prop": null}], [{"surface": "まし", "node_type": "NORMAL",
↪"morphemes": {"conjugated_form": "特殊・マス", "conjugation_type": "連用形",
↪"original_form": "ます", "pos": "助動詞", "prononciation": "マシ", "subclass1":
↪"*", "subclass2": "*", "subclass3": "*", "surface": "まし", "yomi": "マシ"},
↪"geometry": null, "prop": null}], [{"surface": "た", "node_type": "NORMAL",
↪"morphemes": {"conjugated_form": "特殊・タ", "conjugation_type": "基本形",
↪"original_form": "た", "pos": "助動詞", "prononciation": "タ", "subclass1": "*",
 "subclass2": "*", "subclass3": "*", "surface": "た", "yomi": "タ"}, "geometry":␣
↪null, "prop": null}], [{"surface": "。", "node_type": "NORMAL", "morphemes": {
↪"conjugated_form": "*", "conjugation_type": "*", "original_form": "。", "pos":
↪"記号", "prononciation": "。", "subclass1": "句点", "subclass2": "*", "subclass3
↪": "*", "surface": "。", "yomi": "。"}, "geometry": null, "prop": null}]]
```

注釈: ラティス表現では全ての地名語の候補を列挙して返します。analyze_sentence() に住所ジオコーディングの結果も追加します。

analyze_sentence(*sentence*, ***kwargs*)

sentence を解析し、全ての地名語候補を含む Node リストをラティス表現で返します。

パラメータ

sentence (*str*) -- 解析対象の文字列。

戻り値

ラティス表現。形態素ごとに、対応する地名語候補のリストを含むリストです。

戻り値の型

list

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> parser = api.parser.Parser()
>>> for nodes in parser.analyze_sentence('NII は千代田区一ツ橋 2-1-2 にあります。'):
...     [x.simple() for x in nodes]
...
```

(次のページに続く)

(前のページからの続き)

```
['NII(NORMAL)']
[' は (NORMAL)']
["千代田区 (GEOWORD:[' 東京都'])"]
[' 一ツ橋 (NORMAL)']
['2(NORMAL)']
['-(NORMAL)']
['1(NORMAL)']
['-(NORMAL)']
['2(NORMAL)']
[' に (NORMAL)']
[' あり (NORMAL)']
[' ます (NORMAL)']
['。(NORMAL)']
```

check_word(*word*, *filter*)

Word の形態素情報が filter に含まれる全ての key, value と一致しているかどうかを調べます。

パラメータ

- **word** (*dict*) -- チェック対象となる dict オブジェクト。
- **filter** (*dict*) -- チェックする項目と値を持つ dict オブジェクト。

戻り値

全ての項目が一致する場合 True, 一つでも一致しない場合 False。

戻り値の型

bool

get_addresses(*lattice*, *pos*)

ラティス表現の単語列から住所部分を抽出します。

パラメータ

- **lattice** (*list*) -- analyze_sentence() が返すノードのリスト（ラティス表現）。
- **pos** (*int*) -- 住所抽出を開始するリストのインデックス。

戻り値

以下の要素を持つ dict オブジェクトを返します。

address: jageocoder.address.AddressNode

ジオコーディングの結果, 住所ではなかった場合 None。

pos: int 住
所とみなされた形態素ノードの次のインデックス。

戻り値の型
dict

get_surfaces(*lattice*, *pos_from*, *limit*)

ラティス表現の単語列の pos 番目から、 limit で指定した文字数を超える位置までの表記のリストを取得します。

単語の surface と original_form の組み合わせを列挙します。

パラメータ

- **lattice** (*list*) -- analyze_sentence() が返すノードのリスト（ラティス表現）。
- **pos_from** (*int*) -- 表記リストの先頭となるノードのインデックス。
- **limit** (*int*) -- 文字列の長さ（limit を超えたノードを含む）

戻り値 単
語列（単語表記のリスト）

戻り値の型
list

set_jageocoder(*jageocoder*)

この Parser が利用する jageocoder を変更します。

パラメータ
jageocoder (*jageocoder.tree.AddressTree, optional*) -- 利用する住所ジオコーダーを指定します。省略した場合、jageocoder モジュールのデフォルトオブジェクトを利用します。False を指定した場合、ジオコーディング機能を利用しません。

class pygeonlp.api.parser.**ParseError**

ベースクラス: RuntimeError

パージング処理の際に例外が起こると、このクラスが発生します。

14.2.8 pygeonlp.api.scoring module

scoring モジュールは、一通りのジオパージング処理を行なうために必要最小限の、スコアリング処理の実装をまとめています。十分な精度が得られない場合、用途に応じてチューニングを行なうか、別の手法を実装して利用してください。

class pygeonlp.api.scoring.**ScoringClass**(*options=None*)

ベースクラス: object

node_relation_score(*node0*, *node1*)

node クラスのインスタンス node0 と node1 との関係によるスコアを計算するデフォルト実装です。関係が強いノードほど高いスコアを返します。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.scoring import ScoringClass
>>> api.init()
>>> s = ScoringClass()
>>> s.node_relation_score(api.analyze('国会議事堂前')[0][0],
...   api.analyze('永田町')[0][0])
20
```

注釈: スコアは以下のように計算しています。

どちらかのノードが地名語でも住所でもない（NORMAL）場合 0
点

どちらのノードも地名語で、固有名クラスが一致する場合（同クラス）
+10 点

どちらのノードがもう一方の hypernym **にふくまれる場合（親子関係）**
+5 点

両方のノードの hypernym **が** 1 **つ以上一致する場合（兄弟関係）**
+5 点

ノード間の距離（単位は degree**）を計算し、**
0.2 度以下の場合は +5 点、それ以上離れている場合は +1/距離を加算

---

path_score(*path*)

パス表現の結果を一つ受け取り、スコアを計算して返すデフォルト実装です。

パラメータ

path (*list of* Node) -- 解析結果候補のパス表現。

戻り値

与えられたパスを評価したスコアを返します。値が大きいパスほど geoparse() の結果で上位候補として返されます。

戻り値の型

int

注釈: 次のように得点を計算しています。

- 地名語・住所を候補として全く含まない形態素が存在する場合

  + 形態素数 × 1 点（地名語か非地名語か不明な形態素は非地名語を優先）

- 全ての候補の中に住所ノード候補が含まれている場合

  + 住所階層数（' 東京都/千代田区' なら 2）× 10 点

- 全ての候補の中に含まれる地名語ノード候補に、同じ固有名クラスが 2 回以上出現する場合

  + 出現回数 × 10 点

- ある形態素に地名語ノード候補が含まれる場合

  nlookup 先までの形態素に含まれる地名語・住所ノード候補との関係によるスコアを node_relation_score() で計算し加算

14.2.9 pygeonlp.api.service module

class pygeonlp.api.service.**Service**(*db_dir: str | bytes | PathLike | None = None*, *geoword_rules: dict = {}*, ***options*)

ベースクラス: object

文の解析や地名語の検索機能を提供する低水準 API サービスクラスです。

_dict_cache

辞書 ID（整数）をキー、辞書 identifier（文字列）を値にもつマッピングテーブル（キャッシュ用）。

Type

dict

capi_ma

C 実装の拡張形態素解析機能を利用するためのオブジェクト。

Type
pygeonlp.capi オブジェクト

db_dir

このサービスが利用するデータベースディレクトリのパス。

Type
str

__init__(*db_dir: str | bytes | PathLike | None = None*, *geoword_rules: dict = {}*, ***options*)

このサービスが利用する辞書や、各種設定を初期化します。

パラメータ

- **db_dir** (*PathLike, optional*) -- 利用するデータベースディレクトリのパス。省略した場合は `api.get_db_dir()` で決定します。
- **geoword_rules** (*dict, optional*) -- 地名語を解析するルールセットを指定します。
- **options** (*dict, optional*) -- その他の解析オプションを指定します。

注釈: 地名語抽出ルールとして指定できる項目は以下の通りです。

suffix
[list of str] 地名接尾語を定義します。通常、一般名詞などの前に地名がくる場合、地名修飾語として解析されます。例：千代田区立 -> 千代田（名詞・固有名詞・地名語）＋区立（名詞・一般）

地名接尾語として「立」を定義すると、千代田区（名詞・固有名詞・地名語）＋立（名詞・接尾・地名語）と解析されるようになります。地名接尾辞は "立, リツ, リツ" のように表層形、読み、発音をカンマで区切った文字列として定義してください。デフォルト値は ["前, マエ, マエ", "内, ナイ, ナイ", "立, リツ, リツ", "境, サカイ, サカイ", "東, ヒガシ, ヒガシ", "西, ニシ, ニシ", "南, ミナミ, ミナミ", "北, キタ, キタ"] です。

excluded_word
[list of str] 非地名語を定義します。地名解析辞書に登録されている語は地名語として解析されます。例：本部に行く -> 本部（名詞・固有名詞・地名語）＋に（助詞）……しかし特定の語を地名語として解析したくない場合は非地名語として定義してください。デフォルト値は ["本部","一部","月"] です。環境変数 GEONLP_EXCLUDED_WORD でデフォルト値を設定できます。

その他の解析オプションは以下の通りです。

address_class
[str] 住所要素とみなす固有名クラスを正規表現で指定します。たとえば固有名クラスが「都道府県」である地名語から始まる住所表記だけを住所として解析したい（市区町村から始まる場合は無

視したい）場合は r"^都道府県" を指定します。デフォルト値は r"^(都道府県|市区町村|行政地域|居住地名)(/.+|)" です。環境変数 GEONLP_ADRESS_CLASS でデフォルト値を設定できます。

system_dic_dir
: [str] MeCab システム辞書のディレクトリを指定します。省略した場合はデフォルトシステム辞書を利用します。環境変数 GEONLP_MECAB_DIC_DIR でデフォルト値を設定できます。

activateDictionaries(*idlist=None*, *pattern=None*)

指定した辞書を再び利用するようにします。既に利用可能な辞書は指定しなくても利用可能なままになります。新たに利用可能になった辞書のリストを返します。

パラメータ

- **idlist** (*list*) -- 利用する辞書の内部 id または identifier のリスト。
- **pattern** (*str*) -- 利用する辞書の identifier を指定する正規表現。

戻り値

新たに利用可能になった辞書の identifier のリスト。

戻り値の型

list

サンプル

```
>>> from pygeonlp.api.service import Service
>>> service = Service()
>>> sorted(service.disactivateDictionaries(pattern=r'.*'))
['geonlp:geoshape-city', 'geonlp:geoshape-pref', 'geonlp:ksj-station-N02']
>>> [x.get_identifier() for x in service.getActiveDictionaries()]
[]
>>> service.activateDictionaries(pattern=r'ksj-station')
['geonlp:ksj-station-N02']
>>> [x.get_identifier() for x in service.getActiveDictionaries()]
['geonlp:ksj-station-N02']
```

注釈: idlist と pattern を同時に指定した場合、どちらか一方の条件に一致する辞書は利用可能になります。

---

disactivateDictionaries(*idlist=None*, *pattern=None*)

指定した辞書を一時的に除外し、利用しないようにします。既に除外されている辞書は除外されたままになります。新たに除外された辞書のリストを返します。

パラメータ

- **idlist** (*list*) -- 除外する辞書の内部 id または identifier のリスト。
- **pattern** (*str*) -- 除外する辞書の identifier を指定する正規表現。

戻り値

新たに除外された辞書の identifier のリスト。

戻り値の型

list

サンプル

```
>>> from pygeonlp.api.service import Service
>>> service = Service()
>>> service.disactivateDictionaries(pattern=r'geonlp:geoshape')
['geonlp:geoshape-city', 'geonlp:geoshape-pref']
>>> [x.get_identifier() for x in service.getActiveDictionaries()]
['geonlp:ksj-station-N02']
>>> service.disactivateDictionaries(pattern=r'ksj-station')
['geonlp:ksj-station-N02']
>>> [x.get_identifier() for x in service.getActiveDictionaries()]
[]
```

注釈: idlist と pattern を同時に指定した場合、どちらか一方の条件に一致する辞書は除外されます。

---

getActiveClasses()

解析に利用する固有名クラスの一覧を返します。デフォルトは '.*' で、全ての固有名クラスが利用されます。

一時的に特定の固有名クラスだけを解析対象としたい場合、setActiveClasses() で対象クラスを指定できます。

戻り値

利用する固有名クラス（正規表現）のリスト。

戻り値の型

list

サンプル

```
>>> from pygeonlp.api.service import Service
>>> service = Service()
>>> service.getActiveClasses()
['.*']
```

`getActiveDictionaries`()

インストール済み辞書のうち、解析に利用する辞書のメタデータ一覧を返します。デフォルトでは全てのインストール済み辞書を利用します。

一時的に辞書を利用したくない場合、 `disactivateDictionaries()` で除外できます。除外された辞書は `activateDictionaries()` で再び利用可能になります。

戻り値

Metadata インスタンスのリスト。

戻り値の型

list

サンプル

```
>>> from pygeonlp.api.service import Service
>>> service = Service()
>>> sorted([x.get_identifier() for x in service.getActiveDictionaries()])
['geonlp:geoshape-city', 'geonlp:geoshape-pref', 'geonlp:ksj-station-N02']
```

`getWordInfo`(*geolod_id*)

指定した geolod_id を持つ語の情報を返します。id が辞書に存在しない場合は None を返します。

パラメータ

geolod_id (*str or Iterable*) -- 語の ID または ID のリストを返すイテレータ。

戻り値

geolod_id をキー、語の情報を値に持つ dict。

戻り値の型

dict

サンプル

```
>>> from pygeonlp.api.service import Service
>>> service = Service()
>>> service.getWordInfo('Bn4q6d')
{'body': '国会議事堂前', 'dictionary_id': 3, 'entry_id': 'LrGGxY', 'geolod_id':
↪'Bn4q6d', 'hypernym': ['東京地下鉄', '4 号線丸ノ内線'], 'institution_type': '民営
鉄道', 'latitude': '35.674845', 'longitude': '139.74534166666666', 'ne_class': '
鉄道施設/鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅', ''], 'dictionary_
↪identifier': 'geonlp:ksj-station-N02'}
```

ma_parse(*sentence*)

センテンスを形態素解析した結果を MeCab 互換の文字列として返します。

パラメータ

sentence (*str*) -- 解析する文字列。

戻り値

解析結果の改行区切りテキスト。

戻り値の型

str

サンプル

```
>>> from pygeonlp.api.service import Service
>>> service = Service()
>>> print(service.ma_parse('今日は国会議事堂前まで歩きました。'))

今日　　名詞,副詞可能,*,*,*,*,今日,キョウ,キョー
は　　助詞,係助詞,*,*,*,*,は,ハ,ワ
国会議事堂前　　名詞,固有名詞,地名語,Bn4q6d:国会議事堂前駅/cE8W4w:国会議事堂前駅,*,*,
国会議事堂前,,
まで　　助詞,副助詞,*,*,*,*,まで,マデ,マデ
歩き　　動詞,自立,*,*,五段・カ行イ音便,連用形,歩く,アルキ,アルキ
まし　　助動詞,*,*,*,特殊・マス,連用形,ます,マシ,マシ
た　　助動詞,*,*,*,特殊・タ,基本形,た,タ,タ
。　　記号,句点,*,*,*,*,。,。,。
EOS
```

ma_parseNode(*sentence*)

センテンスを形態素解析した結果を MeCab 互換のノード配列として返します。

パラメータ

sentence (*str*) -- 解析する文字列。

戻り値

解析結果のリスト。

戻り値の型

list

サンプル

```
>>> from pygeonlp.api.service import Service
>>> service = Service()
>>> service.ma_parseNode('今日は国会議事堂前まで歩きました。')
[{'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '*', 'pos':
↪'BOS/EOS', 'prononciation': '*', 'subclass1': '*', 'subclass2': '*', 'subclass3
↪': '*', 'surface': '', 'yomi': '*'}, {'conjugated_form': '*', 'conjugation_type
↪': '*', 'original_form': '今日', 'pos': '名詞', 'prononciation': 'キョー',
 'subclass1': '副詞可能', 'subclass2': '*', 'subclass3': '*', 'surface': '今日',
↪'yomi': 'キョウ'}, {'conjugated_form': '*', 'conjugation_type': '*', 'original_
↪form': 'は', 'pos': '助詞', 'prononciation': 'ワ', 'subclass1': '係助詞',
↪'subclass2': '*', 'subclass3': '*', 'surface': 'は', 'yomi': 'ハ'}, {
↪'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '国会議事堂前
', 'pos': '名詞', 'prononciation': '', 'subclass1': '固有名詞', 'subclass2': '地
名語', 'subclass3': 'Bn4q6d:国会議事堂前駅/cE8W4w:国会議事堂前駅', 'surface': '国会
議事堂前', 'yomi': ''}, {'conjugated_form': '*', 'conjugation_type': '*',
↪'original_form': 'まで', 'pos': '助詞', 'prononciation': 'マデ', 'subclass1': '
副助詞', 'subclass2': '*', 'subclass3': '*', 'surface': 'まで', 'yomi': 'マデ'},
↪{'conjugated_form': '五段・カ行イ音便', 'conjugation_type': '連用形', 'original_
↪form': '歩く', 'pos': '動詞', 'prononciation': 'アルキ', 'subclass1': '自立',
↪'subclass2': '*', 'subclass3': '*', 'surface': '歩き', 'yomi': 'アルキ'}, {
↪'conjugated_form': '特殊・マス', 'conjugation_type': '連用形', 'original_form':
↪'ます', 'pos': '助動詞', 'prononciation': 'マシ', 'subclass1': '*', 'subclass2
↪': '*', 'subclass3': '*', 'surface': 'まし', 'yomi': 'マシ'}, {'conjugated_form
↪': '特殊・タ', 'conjugation_type': '基本形', 'original_form': 'た', 'pos': '助
動詞', 'prononciation': 'タ', 'subclass1': '*', 'subclass2': '*', 'subclass3': '*
↪', 'surface': 'た', 'yomi': 'タ'}, {'conjugated_form': '*', 'conjugation_type
↪': '*', 'original_form': '。', 'pos': '記号', 'prononciation': '。', 'subclass1
↪': '句点', 'subclass2': '*', 'subclass3': '*', 'surface': '。', 'yomi': '。'},
↪{'conjugated_form': '*', 'conjugation_type': '*', 'original_form': '*', 'pos':
↪'BOS/EOS', 'prononciation': '*', 'subclass1': '*', 'subclass2': '*', 'subclass3
```

(次のページに続く)

(前のページからの続き)

```
↪': '*', 'surface': '', 'yomi': '*'}]
```

searchWord(*key*)

指定した表記または読みを持つ語の情報を返します。一致する語が辞書に存在しない場合は None を返します。

パラメータ

key (*str*) -- 語の表記または読み。

戻り値

geolod_id をキー、語の情報を値に持つ dict。

戻り値の型

dict

サンプル

```
>>> from pygeonlp.api.service import Service
>>> service = Service()
>>> service.searchWord('国会議事堂前')
{'Bn4q6d': {'body': '国会議事堂前', 'dictionary_id': 3, 'entry_id': 'LrGGxY',
 'geolod_id': 'Bn4q6d', 'hypernym': ['東京地下鉄', '4 号線丸ノ内線'], 'institution_
↪type': '民営鉄道', 'latitude': '35.674845', 'longitude': '139.74534166666666',
 'ne_class': '鉄道施設/鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅', ''],
 'dictionary_identifier': 'geonlp:ksj-station-N02'}, 'cE8W4w': {'body': '国会議事
堂前', 'dictionary_id': 3, 'entry_id': '4NFELa', 'geolod_id': 'cE8W4w', 'hypernym
↪': ['東京地下鉄', '9 号線千代田線'], 'institution_type': '民営鉄道', 'latitude':
↪'35.673543333333335', 'longitude': '139.74305333333334', 'ne_class': '鉄道施設/
↪鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅', ''], 'dictionary_
↪identifier': 'geonlp:ksj-station-N02'}}
```

setActiveClasses(*patterns=None*)

解析対象とする固有名クラスの正規表現リストを指定します。いずれかの正規表現に一致する固有名クラスは解析対象となります。

'-' から始まる場合、その正規表現に一致する固有名クラスは対象外となります。

パラメータ

patterns (*list, optional*) -- 解析対象とする固有名クラス（str）のリスト。省略した場合 ['.*'] （全固有名クラス）を対象とします。

サンプル

```
>>> from pygeonlp.api.service import Service
>>> service = Service()
>>> service.getActiveClasses()
['.*']
>>> service.searchWord('東京都')
{'QknGsa': {...'dictionary_identifier': 'geonlp:geoshape-pref'}}
>>> service.setActiveClasses(['.*', '-都道府県'])
>>> service.searchWord('東京都')
{}
>>> service.setActiveClasses()
>>> service.getActiveClasses()
['.*']
```

setActiveDictionaries(*idlist=None*, *pattern=None*)

インストール済み辞書のうち、解析に利用する辞書を指定します。

パラメータ

- **idlist** (*list*, *optional*) -- 利用する辞書の id または identifier のリスト。
- **pattern** (*str*, *optional*) -- 利用する辞書の identifier の正規表現。

サンプル

```
>>> from pygeonlp.api.service import Service
>>> service = Service()
>>> service.setActiveDictionaries(pattern=r'geonlp:geoshape')
>>> sorted([x.get_identifier() for x in service.getActiveDictionaries()])
['geonlp:geoshape-city', 'geonlp:geoshape-pref']
```

注釈: idlist と pattern のどちらかは指定する必要があります。

---

class pygeonlp.api.service.**ServiceError**

ベースクラス: RuntimeError

サービス実行時に例外が起こると、このクラスが発生します。

14.2.10 pygeonlp.api.spatial_filter module

14.2.11 pygeonlp.api.temporal_filter module

class pygeonlp.api.temporal_filter.**TemporalFilter**(*date_from*, *date_to=None*, ***kwargs*)

ベースクラス: *Filter*

時間フィルタの基底クラス。

時間フィルタは、形態素に対応する地名語ノード候補のうち、条件に一致するものを残し、一致しないものを削除します。全ての候補が条件に一致しない場合、品詞情報として固有名詞を持つ一般ノードを一つ作成して置き換えます。

時間フィルタでは、地名語ノードの prop.valid_from および prop.valid_to をそのノードが存在する期間とします。

valid_from が設定されていない場合
: 無限の過去から存在していたとみなします。

valid_to が設定されていない場合
: 無限の未来まで存在しているとみなします。

duration

期間の開始日と終了日のタプル。

Type
: (datetime.date, datetime.date)

__init__(*date_from*, *date_to=None*, ***kwargs*)

パラメータ

- **date_from** (*str, datetime.date, datetime.datetime*) -- 期間の開始日時。
- **date_to** (*str, datetime.date, datetime.datetime, optional*) -- 期間の終了日時。

注釈: date_to を省略した場合、date_from が表す一時点が期間になります。たとえば TimeExistsFilter('2001-01-01') は、2001 年 1 月 1 日時点に存在していた地名語ノードを残すフィルタになります。

---

classmethod duration_from_candidate(*candidate*)

候補ノードの valid_from, valid_to から有効期間を表す tuple を取得する。valid_from, valid_to が含まれていない場合は None を返す。

classmethod get_date_from_isostr(*datestr*)

ISO フォーマット、または類似したフォーマットから date に相当する部分を取得します。

パラメータ

datestr (*str*) -- 日付または日時文字列。

戻り値

日付部分を表す date オブジェクト。

戻り値の型

datetime.date

classmethod get_duration_from_dates(*datestr0*, *datestr1*)

2 つの日付文字列から期間を表す tuple を取得します。

パラメータ

- **datestr0** (*str*, *datetime.date*, *datetime.datetime*) -- 期間の開始日時を指定します。
- **datestr1** (*str*, *datetime.date*, *datetime.datetime*) -- 期間の終了日時を指定します。

戻り値

期間の開始日時と終了日時を表す datetime.date オブジェクトの tuple を返します。

戻り値の型

tuple

class pygeonlp.api.temporal_filter.**TimeExistsFilter**(*date_from*, *date_to=None*, ***kwargs*)

ベースクラス: *TemporalFilter*

指定した時間範囲に存在していれば合格とするフィルタを作成します。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.temporal_filter import TimeExistsFilter
>>> api.init()
>>> tefilter = TimeExistsFilter('2000-01-01', '2001-01-01')
>>> lattice = api.analyze('田無市と保谷市は2001年1月21日に合併して西東京市になりました。
')
>>> for nodes in lattice:
...     [x.simple() for x in nodes]
...
["田無市(GEOWORD:['東京都'])"]
```

(次のページに続く)

(前のページからの続き)

```
['と(NORMAL)']
["保谷市(GEOWORD:['東京都'])"]
['は(NORMAL)']
['2001(NORMAL)']
['年(NORMAL)']
['1(NORMAL)']
['月(NORMAL)']
['21(NORMAL)']
['日(NORMAL)']
['に(NORMAL)']
['合併(NORMAL)']
['し(NORMAL)']
['て(NORMAL)']
["西東京市(GEOWORD:['東京都'])"]
['に(NORMAL)']
['なり(NORMAL)']
['まし(NORMAL)']
['た(NORMAL)']
['。(NORMAL)']
>>> lattice_filtered = tefilter(lattice)
>>> # フィルタを適用すると「西東京市」が範囲外なので地名語候補から除外されます
>>> for nodes in lattice_filtered:
...     [x.simple() for x in nodes]
...
["田無市(GEOWORD:['東京都'])"]
['と(NORMAL)']
["保谷市(GEOWORD:['東京都'])"]
['は(NORMAL)']
['2001(NORMAL)']
['年(NORMAL)']
['1(NORMAL)']
['月(NORMAL)']
['21(NORMAL)']
['日(NORMAL)']
['に(NORMAL)']
['合併(NORMAL)']
['し(NORMAL)']
['て(NORMAL)']
['西東京市(NORMAL)']
```

(次のページに続く)

(前のページからの続き)

```
['に (NORMAL)']
['なり (NORMAL)']
['まし (NORMAL)']
['た (NORMAL)']
['。 (NORMAL)']
```

注釈: このフィルタは TimeOverlapsFilter と同じです。

filter_func(*candidate*)

時間範囲に存在していれば True を返します。

パラメータ

candidate (pygeonlp.api.node.Node) -- 検査対象となるノードオブジェクト。

戻り値

フィルタの条件を満たせば True, 満たさなければ False を返します。

戻り値の型

bool

class pygeonlp.api.temporal_filter.**TimeBeforeFilter**(*date_from*, *date_to=None*, ***kwargs*)

ベースクラス: *TemporalFilter*

指定した時間範囲の開始時より前に存在していれば合格とするフィルタを返します。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.temporal_filter import TimeBeforeFilter
>>> api.init()
>>> tbfilter = TimeBeforeFilter('2000-01-01')
>>> lattice = api.analyze('田無市と保谷市は2001年1月21日に合併して西東京市になりました。
')
>>> for nodes in lattice:
...     [x.simple() for x in nodes]
...
["田無市 (GEOWORD:['東京都'])"]
['と (NORMAL)']
["保谷市 (GEOWORD:['東京都'])"]
```

(次のページに続く)

(前のページからの続き)

```
[' は (NORMAL)']
['2001(NORMAL)']
[' 年 (NORMAL)']
['1(NORMAL)']
[' 月 (NORMAL)']
['21(NORMAL)']
[' 日 (NORMAL)']
[' に (NORMAL)']
[' 合併 (NORMAL)']
[' し (NORMAL)']
[' て (NORMAL)']
["西東京市 (GEOWORD:[' 東京都'])"]
[' に (NORMAL)']
[' なり (NORMAL)']
[' まし (NORMAL)']
[' た (NORMAL)']
['。(NORMAL)']
>>> lattice_filtered = tbfilter(lattice)
>>> # フィルタを適用すると「西東京市」が範囲外なので地名語候補から除外されます
>>> for nodes in lattice_filtered:
...     [x.simple() for x in nodes]
...
["田無市 (GEOWORD:[' 東京都'])"]
[' と (NORMAL)']
["保谷市 (GEOWORD:[' 東京都'])"]
[' は (NORMAL)']
['2001(NORMAL)']
[' 年 (NORMAL)']
['1(NORMAL)']
[' 月 (NORMAL)']
['21(NORMAL)']
[' 日 (NORMAL)']
[' に (NORMAL)']
[' 合併 (NORMAL)']
[' し (NORMAL)']
[' て (NORMAL)']
[' 西東京市 (NORMAL)']
[' に (NORMAL)']
[' なり (NORMAL)']
```

(次のページに続く)

(前のページからの続き)

```
['まし(NORMAL)']
['た(NORMAL)']
['。(NORMAL)']
```

filter_func(*candidate*)

時間範囲の開始時より前に存在していれば True を返します。

パラメータ

candidate (pygeonlp.api.node.Node) -- 検査対象となるノードオブジェクト。

戻り値

フィルタの条件を満たせば True, 満たさなければ False を返します。

戻り値の型

bool

class pygeonlp.api.temporal_filter.**TimeAfterFilter**(*date_from*, *date_to=None*, ***kwargs*)

ベースクラス: *TemporalFilter*

指定した時間範囲の終了時より後に存在していれば合格とするフィルタを作成します。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.temporal_filter import TimeAfterFilter
>>> api.init()
>>> tafilter = TimeAfterFilter('2001-01-22')
>>> lattice = api.analyze('田無市と保谷市は2001年1月21日に合併して西東京市になりました。
')
>>> for nodes in lattice:
...     [x.simple() for x in nodes]
...
["田無市(GEOWORD:['東京都'])"]
['と(NORMAL)']
["保谷市(GEOWORD:['東京都'])"]
['は(NORMAL)']
['2001(NORMAL)']
['年(NORMAL)']
['1(NORMAL)']
['月(NORMAL)']
['21(NORMAL)']
```

(次のページに続く)

(前のページからの続き)

```
['日(NORMAL)']
['に(NORMAL)']
['合併(NORMAL)']
['し(NORMAL)']
['て(NORMAL)']
["西東京市(GEOWORD:['東京都'])"]
['に(NORMAL)']
['なり(NORMAL)']
['まし(NORMAL)']
['た(NORMAL)']
['。(NORMAL)']
>>> lattice_filtered = tafilter(lattice)
>>> # フィルタを適用すると「田無市」「保谷市」が範囲外なので地名語候補から除外されます
>>> for nodes in lattice_filtered:
...     [x.simple() for x in nodes]
...
['田無市(NORMAL)']
['と(NORMAL)']
['保谷市(NORMAL)']
['は(NORMAL)']
['2001(NORMAL)']
['年(NORMAL)']
['1(NORMAL)']
['月(NORMAL)']
['21(NORMAL)']
['日(NORMAL)']
['に(NORMAL)']
['合併(NORMAL)']
['し(NORMAL)']
['て(NORMAL)']
["西東京市(GEOWORD:['東京都'])"]
['に(NORMAL)']
['なり(NORMAL)']
['まし(NORMAL)']
['た(NORMAL)']
['。(NORMAL)']
```

filter_func(*candidate*)

時間範囲の開始時より前に存在していれば True を返します。

パラメータ

candidate (pygeonlp.api.node.Node) -- 検査対象となるノードオブジェクト。

戻り値

フィルタの条件を満たせば True, 満たさなければ False を返します。

戻り値の型

bool

class pygeonlp.api.temporal_filter.**TimeOverlapsFilter**(*date_from*, *date_to=None*, ***kwargs*)

ベースクラス: *TemporalFilter*

指定した時間範囲内に存在していれば合格とするフィルタを作成します。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.temporal_filter import TimeOverlapsFilter
>>> api.init()
>>> tofilter = TimeOverlapsFilter('2001-01-01', '2001-02-01')
>>> lattice = api.analyze('田無市と保谷市は2001年1月21日に合併して西東京市になりました。
')
>>> for nodes in lattice:
...     [x.simple() for x in nodes]
...
["田無市(GEOWORD:['東京都'])"]
['と(NORMAL)']
["保谷市(GEOWORD:['東京都'])"]
['は(NORMAL)']
['2001(NORMAL)']
['年(NORMAL)']
['1(NORMAL)']
['月(NORMAL)']
['21(NORMAL)']
['日(NORMAL)']
['に(NORMAL)']
['合併(NORMAL)']
['し(NORMAL)']
['て(NORMAL)']
["西東京市(GEOWORD:['東京都'])"]
['に(NORMAL)']
```

(次のページに続く)

(前のページからの続き)

```
['なり (NORMAL)']
['まし (NORMAL)']
['た (NORMAL)']
['。 (NORMAL)']
>>> lattice_filtered = tofilter(lattice)
>>> # フィルタを適用しても、「田無市」「保谷市」は 2001 年 1 月 1 日より後まで存在し、
>>> # 「西東京市」は 2001 年 2 月 1 日より前から存在するので、いずれもそのまま残ります
>>> for nodes in lattice_filtered:
...     [x.simple() for x in nodes]
...
["田無市 (GEOWORD:['東京都'])"]
['と (NORMAL)']
["保谷市 (GEOWORD:['東京都'])"]
['は (NORMAL)']
['2001(NORMAL)']
['年 (NORMAL)']
['1(NORMAL)']
['月 (NORMAL)']
['21(NORMAL)']
['日 (NORMAL)']
['に (NORMAL)']
['合併 (NORMAL)']
['し (NORMAL)']
['て (NORMAL)']
["西東京市 (GEOWORD:['東京都'])"]
['に (NORMAL)']
['なり (NORMAL)']
['まし (NORMAL)']
['た (NORMAL)']
['。 (NORMAL)']
```

注釈: このフィルタは TimeExistsFilter と同じです。

filter_func(*candidate*)

時間範囲の開始時より後、または終了時より前に存在していれば True を返します。

パラメータ

candidate (pygeonlp.api.node.Node) -- 検査対象となるノードオブジェクト。

戻り値

フィルタの条件を満たせば True, 満たさなければ False を返します。

戻り値の型

bool

class pygeonlp.api.temporal_filter.**TimeCoversFilter**(*date_from*, *date_to=None*, ***kwargs*)

ベースクラス: *TemporalFilter*

指定した時間範囲の開始時から終了時まで存在していれば合格とするフィルタを作成します。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.temporal_filter import TimeCoversFilter
>>> api.init()
>>> tcfilter = TimeCoversFilter('2001-01-01', '2001-02-01')
>>> lattice = api.analyze('田無市と保谷市は2001年1月21日に合併して西東京市になりました。
')
>>> for nodes in lattice:
...     [x.simple() for x in nodes]
...
["田無市(GEOWORD:['東京都'])"]
['と(NORMAL)']
["保谷市(GEOWORD:['東京都'])"]
['は(NORMAL)']
['2001(NORMAL)']
['年(NORMAL)']
['1(NORMAL)']
['月(NORMAL)']
['21(NORMAL)']
['日(NORMAL)']
['に(NORMAL)']
['合併(NORMAL)']
['し(NORMAL)']
['て(NORMAL)']
["西東京市(GEOWORD:['東京都'])"]
['に(NORMAL)']
['なり(NORMAL)']
['まし(NORMAL)']
['た(NORMAL)']
```

(次のページに続く)

(前のページからの続き)

```
['。(NORMAL)']
>>> lattice_filtered = tcfilter(lattice)
>>> # フィルタを適用すると、「田無市」「保谷市」は 2001 年 2 月 1 日より前に消滅し、
>>> # 「西東京市」は 2001 年 1 月 1 日より前から存在するので、地名語候補から除外されます。
>>> for nodes in lattice_filtered:
...     [x.simple() for x in nodes]
...
['田無市(NORMAL)']
['と(NORMAL)']
['保谷市(NORMAL)']
['は(NORMAL)']
['2001(NORMAL)']
['年(NORMAL)']
['1(NORMAL)']
['月(NORMAL)']
['21(NORMAL)']
['日(NORMAL)']
['に(NORMAL)']
['合併(NORMAL)']
['し(NORMAL)']
['て(NORMAL)']
['西東京市(NORMAL)']
['に(NORMAL)']
['なり(NORMAL)']
['まし(NORMAL)']
['た(NORMAL)']
['。(NORMAL)']
```

filter_func(*candidate*)

時間範囲の開始時より前、かつ、終了時より後に存在していれば True を返します。

パラメータ

candidate (pygeonlp.api.node.Node) -- 検査対象となるノードオブジェクト。

戻り値

フィルタの条件を満たせば True, 満たさなければ False を返します。

戻り値の型

bool

14.2.12 pygeonlp.api.workflow module

class pygeonlp.api.workflow.**Workflow**(*db_dir=None*, *address_regex=None*, *jageocoder=None*, *scoring_class=None*, *scoring_options=None*, *filters=None*, ***options*)

ベースクラス: object

一連のジオパース処理を行なうワークフロー。

parser

テキストを解析し、地名語を埋め込む Parser のインスタンス。Parser はテキスト（文字列）からラティス表現を作成します。

Type

pygeonlp.parser.Parser

filters

ラティス表現に含まれる候補に対し、条件による選別を行い、候補を絞り込む Filter インスタンスのリスト。リストの順番に Filter を適用します。

Type

list of pygeonlp.filter.Filter

evaluator

ラティス表現から順位付きパス表現を作成する Evaluator のインスタンス。

Type

pygeonlp.linker.Evaluator

__init__(*db_dir=None*, *address_regex=None*, *jageocoder=None*, *scoring_class=None*, *scoring_options=None*, *filters=None*, ***options*)

パーザを初期化します。

パラメータ

- **db_dir** (*PathLike, optional*) -- データベースディレクトリ。省略した場合は api.init.get_db_dir() が返す値を利用します。
- **address_regex** (*str, optional*) -- 住所表記の開始とみなす地名語の固有名クラスを表す正規表現。省略した場合、r'^(**都道府県**|**市区町村**|**行政地域**|**居住地名**)(/.+|)' を利用します。
- **jageocoder** (*jageocoder.tree.AddressTree, optional*) -- 利用する住所ジオコーダーを指定します。省略した場合、jageocoder モジュールのデフォルトオブジェクトを利用します。False を指定した場合、ジオコーディング機能を利用しません。

- **scoring_class** (*class, optional*) -- パスのスコアとノード間のスコアを計算する関数を持つスコアリングクラス。指定しない場合、pygeonlp.api.scoring モジュール内の ScoringClass が利用されます。
- **scoring_options** (*any, optional*) -- スコアリングクラスの初期化に渡すオプションパラメータ。
- **filters** (*list*) -- 適用するフィルタオブジェクトのリスト。省略した場合は InputBasedFilter がセットされます。

activateDictionaries(*idlist=None*, *pattern=None*)

disactivateDictionaries(*idlist=None*, *pattern=None*)

geoparse(*sentence: str*)

文を解析して GeoJSON Feature 形式に変換可能な dict のリストを返します。

パラメータ

sentence (*str*) -- 解析する文字列

戻り値

GeoJSON Feature 形式に変換可能な dict のリスト。

戻り値の型

list

サンプル

```
>>> import pygeonlp.api as api
>>> api.init()
>>> workflow = api.workflow.Workflow()
>>> workflow.geoparse('国会議事堂前まで歩きました。')
[{'type': 'Feature', 'geometry': {'type': 'Point', 'coordinates': [139.
↪74305333333334, 35.67354333333335]}, 'properties': {'surface': '国会議事堂前',
↪'node_type': 'GEOWORD', 'morphemes': {'conjugated_form': '*', 'conjugation_type
↪': '*', 'original_form': '国会議事堂前', 'pos': '名詞', 'prononciation': '',
 'subclass1': '固有名詞', 'subclass2': '地名語', 'subclass3': 'cE8W4w:国会議事堂前
駅', 'surface': '国会議事堂前', 'yomi': ''}, 'geoword_properties': {'body': '国会
議事堂前', 'dictionary_id': 3, 'entry_id': '4NFELa', 'geolod_id': 'cE8W4w',
↪'hypernym': ['東京地下鉄', '9 号線千代田線'], 'institution_type': '民営鉄道',
↪'latitude': '35.67354333333335', 'longitude': '139.74305333333334', 'ne_class
↪': '鉄道施設/鉄道駅', 'railway_class': '普通鉄道', 'suffix': ['駅', ''],
↪'dictionary_identifier': 'geonlp:ksj-station-N02'}}}, {'type': 'Feature',
```

(次のページに続く)

(前のページからの続き)

```
↪'geometry': None, 'properties': {'surface': 'まで', 'node_type': 'NORMAL',
 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_form':
↪'まで', 'pos': '助詞', 'prononciation': 'マデ', 'subclass1': '副助詞',
 'subclass2': '*', 'subclass3': '*', 'surface': 'まで', 'yomi': 'マデ'}}}, {'type
↪': 'Feature', 'geometry': None, 'properties': {'surface': '歩き', 'node_type':
↪'NORMAL', 'morphemes': {'conjugated_form': '五段・カ行イ音便', 'conjugation_type
↪': '連用形', 'original_form': '歩く', 'pos': '動詞', 'prononciation': 'アルキ',
 'subclass1': '自立', 'subclass2': '*', 'subclass3': '*', 'surface': '歩き',
↪'yomi': 'アルキ'}}}, {'type': 'Feature', 'geometry': None, 'properties': {
↪'surface': 'まし', 'node_type': 'NORMAL', 'morphemes': {'conjugated_form': '特
殊・マス', 'conjugation_type': '連用形', 'original_form': 'ます', 'pos': '助動詞',
 'prononciation': 'マシ', 'subclass1': '*', 'subclass2': '*', 'subclass3': '*',
↪'surface': 'まし', 'yomi': 'マシ'}}}, {'type': 'Feature', 'geometry': None,
↪'properties': {'surface': 'た', 'node_type': 'NORMAL', 'morphemes': {
↪'conjugated_form': '特殊・タ', 'conjugation_type': '基本形', 'original_form': '
た', 'pos': '助動詞', 'prononciation': 'タ', 'subclass1': '*', 'subclass2': '*',
↪'subclass3': '*', 'surface': 'た', 'yomi': 'タ'}}}, {'type': 'Feature',
↪'geometry': None, 'properties': {'surface': '。', 'node_type': 'NORMAL',
 'morphemes': {'conjugated_form': '*', 'conjugation_type': '*', 'original_form':
↪'。', 'pos': '記号', 'prononciation': '。', 'subclass1': '句点', 'subclass2':
↪'*', 'subclass3': '*', 'surface': '。', 'yomi': '。'}}}]
```

getActiveClasses()

getActiveDictionaries()

get_processible_lattice_part(*lattice*)

組み合わせの候補数が MAX_COMBINATIONS 未満になるようにラティスの先頭部分から区切りの良い一部分を抽出するジェネレータ。

パラメータ

lattice (*list*) -- 入力となるラティス表現。

戻り値

先頭部分から切り出した部分的なラティス表現。

戻り値の型

list

注釈: この関数はジェネレータなので yield で返します。

入力テキストの区切り文字が事前に分かっている場合や、途中で分割されたくない場合には、意図した位置で分割するようにオーバーライドしてください。

---

setActiveClasses(*patterns=None*)

setActiveDictionaries(*idlist=None*, *pattern=None*)

class pygeonlp.api.workflow.**WorkflowError**

ベースクラス: RuntimeError

ワークフロー処理の際に例外が起こると、このクラスが発生します。

14.2.13 pygeonlp.api.devtool module

pygeonlp.api.devtool.**pp_geojson**(*geojson_list*, *indent=2*, *file=None*)

geoparse() の結果（GeoJSON 互換）を pretty print します。

パラメータ

- **geojson_list** (*list*) -- geoparse() 結果の GeoJSON 互換 dict のリスト。
- **indent** (*int, optional*) -- インデント幅。デフォルトは 2 です。
- **file** (*file descriptor, optional*) -- 出力先のファイルデスクリプタ。デフォルトは None です。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.devtool import pp_geojson
>>> api.init()
>>> pp_geojson(api.geoparse('アメリカ大使館：港区赤坂'))
アメリカ大使館 ： 【港区赤坂:['東京都', '港区', '赤坂']】 EOS
```

pygeonlp.api.devtool.**pp_lattice**(*lattice*, *indent=2*, *file=None*)

ラティス表現のデータを pretty print します。

パラメータ

- **lattice** (*list*) -- 解析結果のラティス表現。
- **indent** (*int, optional*) -- インデント幅。デフォルトは 2 です。

- **file**(*file descriptor, optional*) -- 出力先のファイルデスクリプタ。デフォルトは None です。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.devtool import pp_lattice
>>> api.init()
>>> parser = api.parser.Parser(jageocoder=True)
>>> lattice = parser.analyze_sentence('アメリカ大使館：港区赤坂1-10-5')
>>> pp_lattice(lattice)
#0:'アメリカ大使館'
  アメリカ大使館(NORMAL)
#1:'：'
  ：(NORMAL)
#2:'港区'
  港区(GEOWORD:['東京都'])
  港区(GEOWORD:['愛知県', '名古屋市'])
  港区(GEOWORD:['大阪府', '大阪市'])
#3:'赤坂'
  赤坂(GEOWORD:['上毛電気鉄道', '上毛線'])
  赤坂(GEOWORD:['東京地下鉄', '9号線千代田線'])
  赤坂(GEOWORD:['富士急行', '大月線'])
  赤坂(GEOWORD:['福岡市', '1号線(空港線)'])
#4:'1'
  1(NORMAL)
#5:'-'
  -(NORMAL)
#6:'10'
  10(NORMAL)
#7:'-'
  -(NORMAL)
#8:'5'
  5(NORMAL)
>>> lattice_address = parser.add_address_candidates(lattice, True)
>>> pp_lattice(lattice_address)
#0:'アメリカ大使館'
  アメリカ大使館(NORMAL)
#1:'：'
  ：(NORMAL)
```

(次のページに続く)

(前のページからの続き)

```
#2:'港区'
  港区(GEOWORD:['東京都'])
  港区(GEOWORD:['愛知県', '名古屋市'])
  港区(GEOWORD:['大阪府', '大阪市'])
  港区赤坂1-10-(ADDRESS:東京都/港区/赤坂/一丁目/10番)[6]
#3:'赤坂'
  赤坂(GEOWORD:['上毛電気鉄道', '上毛線'])
  赤坂(GEOWORD:['東京地下鉄', '9号線千代田線'])
  赤坂(GEOWORD:['富士急行', '大月線'])
  赤坂(GEOWORD:['福岡市', '1号線(空港線)'])
#4:'1'
  1(NORMAL)
#5:'-'
  -(NORMAL)
#6:'10'
  10(NORMAL)
#7:'-'
  -(NORMAL)
#8:'5'
  5(NORMAL)
>>> lattice_address_compact = parser.add_address_candidates(lattice)
>>> pp_lattice(lattice_address_compact)
#0:'アメリカ大使館'
  アメリカ大使館(NORMAL)
#1:'：'
  ：(NORMAL)
#2:'港区赤坂1-10-'
  港区赤坂1-10-(ADDRESS:東京都/港区/赤坂/一丁目/10番)[6]
#3:'5'
  5(NORMAL)
```

pygeonlp.api.devtool.**pp_mecab**(*geojson_list*, *file=None*)

geoparse() の結果（GeoJSON 互換）を mecab 類似書式で出力します。

パラメータ

- **geojson_list** (*list*) -- geoparse() 結果の GeoJSON 互換 dict のリスト。
- **file** (*file descriptor, optional*) -- 出力先のファイルデスクリプタ。デフォルトは None です。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.devtool import pp_mecab
>>> pp_mecab(api.geoparse('目黒駅は品川区にあります。'))
目黒駅　名詞,固有名詞,地名語,Xy26iV:目黒駅,*,*,目黒駅,,鉄道施設/鉄道駅,Xy26iV,目黒駅,
↪139.71566,35.632485
は　助詞,係助詞,*,*,*,*,は,ハ,ワ
品川区　名詞,固有名詞,地名語,kEAYBl:品川区,*,*,品川区,,市区町村,kEAYBl,品川区,139.
↪73025000,35.60906600
に　助詞,格助詞,一般,*,*,*,に,ニ,ニ
あり　動詞,自立,*,*,連用形,五段・ラ行,ある,アリ,アリ
ます　助動詞,*,*,*,基本形,特殊・マス,ます,マス,マス
。　記号,句点,*,*,*,*,。,。,。
EOS
```

pygeonlp.api.devtool.**pp_path**(*path*, *indent=2*, *file=None*)

パス表現のデータを pretty print します。

パラメータ

- **path** (*list*) -- 解析結果のパス表現。
- **indent** (*int, optional*) -- インデント幅。デフォルトは 2 です。
- **file** (*file descriptor, optional*) -- 出力先のファイルデスクリプタ。デフォルトは None です。

サンプル

```
>>> import pygeonlp.api as api
>>> from pygeonlp.api.linker import LinkedResults
>>> from pygeonlp.api.devtool import pp_path
>>> api.init(jageocoder=False)
>>> lattice = api.analyze('アメリカ大使館：港区赤坂 1-10-5')
>>> for path in LinkedResults(lattice):
...   pp_path(path)
...
[
  #0:アメリカ大使館 (NORMAL)
  #1:：(NORMAL)
  #2:港区 (GEOWORD:['東京都'])
```

(次のページに続く)

(前のページからの続き)

```
  #3: 赤坂 (GEOWORD:[' 上毛電気鉄道', ' 上毛線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
[
  #0: アメリカ大使館 (NORMAL)
  #1:：(NORMAL)
  #2: 港区 (GEOWORD:[' 東京都'])
  #3: 赤坂 (GEOWORD:[' 東京地下鉄', '9 号線千代田線'])
  #4:1(NORMAL)
  #5:-(NORMAL)
  #6:10(NORMAL)
  #7:-(NORMAL)
  #8:5(NORMAL)
]
...
```

第 15 章

pygeonlp.webapi package

pygeonlp.webapi パッケージを構成するモジュールおよびメソッドのリファレンス。

15.1 webapi.app モジュール

Python モジュール索引

p

索引