usadress Documentation

Release 0.5.4

Cathy Deng, Forest Gregg

Jul 17, 2018

Contents

usaddress is a python library for parsing unstructured address strings into address components, using advanced NLP methods.

CHAPTER 1

Installation

```
pip install usaddress
```

CHAPTER 2

Usage

The `parse` method will split your address string into components, and label each component.

```
>>> import usaddress
>>> usaddress.parse('Robie House, 5757 South Woodlawn Avenue, Chicago, IL 60637')
[('Robie', 'BuildingName'),
('House,', 'BuildingName'),
('5757', 'AddressNumber'),
('South', 'StreetNamePreDirectional'),
('Woodlawn', 'StreetName'),
('Avenue,', 'StreetNamePostType'),
('Chicago,', 'PlaceName'),
('IL', 'StateName'),
('60637', 'ZipCode')]
```

The `tag` method will try to be a little smarter - it will merge consecutive components & strip commas, as well as return an addr

```
>>> import usaddress
>>> usaddress.tag('Robie House, 5757 South Woodlawn Avenue, Chicago, IL 60637')
(OrderedDict([
   ('BuildingName', 'Robie House'),
   ('AddressNumber', '5757'),
   ('StreetNamePreDirectional', 'South'),
   ('StreetName', 'Woodlawn'),
   ('StreetNamePostType', 'Avenue'),
   ('PlaceName', 'Chicago'),
   ('StateName', 'IL'),
   ('ZipCode', '60637')]),
'Street Address')
>>> usaddress.tag('State & Lake, Chicago')
(OrderedDict([
   ('StreetName', 'State'),
   ('IntersectionSeparator', '&'),
   ('SecondStreetName', 'Lake'),
```

(continues on next page)

(continued from previous page)

```
    ('PlaceName', 'Chicago')]),
'Intersection')
>>> usaddress.tag('P.O. Box 123, Chicago, IL')
(OrderedDict([
    ('USPSBoxType', 'P.O. Box'),
    ('USPSBoxID', '123'),
    ('PlaceName', 'Chicago'),
    ('StateName', 'IL')]),
'PO Box')
```

Because the `tag` method returns an OrderedDict with labels as keys, it will throw a `RepeatedLabelError` error when multiple areas of an address have the same label, and thus can't be concatenated. When `RepeatedLabelError` is raised, it is likely that either (1) the input string is not a valid address, or (2) some tokens were labeled incorrectly.

`RepeatedLabelError` has the attributes `original_string` (the input string) and `parsed_string` (the output of the pa

```
try:
    tagged_address, address_type = usaddress.tag(string)
except usaddress.RepeatedLabelError as e :
    some_special_instructions(e.parsed_string, e.original_string)
```

It is also possible to pass a mapping dict to the `tag` method to remap the labels to your own format. For example:

```
>>> import usaddress
>>> address = 'Robie House, 5757 South Woodlawn Avenue, Chicago, IL 60637'
>>> usaddress.tag(address, tag_mapping={
    'Recipient': 'recipient',
    'AddressNumber': 'address1',
    'AddressNumberPrefix': 'address1',
    'AddressNumberSuffix': 'address1',
    'StreetName': 'address1',
    'StreetNamePreDirectional': 'address1',
    'StreetNamePreModifier': 'address1',
    'StreetNamePreType': 'address1',
    'StreetNamePostDirectional': 'address1',
    'StreetNamePostModifier': 'address1',
    'StreetNamePostType': 'address1',
    'CornerOf': 'address1',
    'IntersectionSeparator': 'address1',
    'LandmarkName': 'address1',
    'USPSBoxGroupID': 'address1',
    'USPSBoxGroupType': 'address1',
    'USPSBoxID': 'address1',
    'USPSBoxType': 'address1',
    'BuildingName': 'address2',
    'OccupancyType': 'address2',
    'OccupancyIdentifier': 'address2',
    'SubaddressIdentifier': 'address2',
    'SubaddressType': 'address2',
    'PlaceName': 'city',
    'StateName': 'state',
    'ZipCode': 'zip_code',
})
(OrderedDict([
    ('address2', u'Robie House'),
```

(continues on next page)

(continued from previous page)

```
   ('address1', u'5757 South Woodlawn Avenue'),
   ('city', u'Chicago'),
   ('state', u'IL'),
   ('zip_code', u'60637')]
),
'Street Address')
```

CHAPTER 3

Details

The address components are based upon the United States Thoroughfare, Landmark, and Postal Address Data Standard, and usaddress knows about the following types of components:

- **AddressNumber** - address number
- **AddressNumberPrefix** - a modifier before an address number, e.g. ‘Mile’, ‘#’
- **AddressNumberSuffix** - a modifier after an address number, e.g ‘B’, ‘1/2’
- **BuildingName** - the name of a building, e.g. ‘Atlanta Financial Center’
- **CornerOf** - words indicating that an address is a corner, e.g. ‘Junction’, ‘corner of’
- **IntersectionSeparator** - a conjunction connecting parts of an intersection, e.g. ‘and’, ‘&’
- **LandmarkName** - the name of a landmark, e.g. ‘Wrigley Field’, ‘Union Station’
- **NotAddress** - a non-address component that doesn’t refer to a recipient
- **OccupancyType** - a type of occupancy within a building, e.g. ‘Suite’, ‘Apt’, ‘Floor’
- **OccupancyIdentifier** - the identifier of an occupancy, often a number or letter
- **PlaceName** - city
- **Recipient** - a non-address recipient, e.g. the name of a person/organization
- **StateName** - state
- **StreetName** - street name, excluding type & direction
- **StreetNamePreDirectional** - a direction before a street name, e.g. ‘North’, ‘S’
- **StreetNamePreModifier** - a modifier before a street name that is not a direction, e.g. ‘Old’
- **StreetNamePreType** - a street type that comes before a street name, e.g. ‘Route’, ‘Ave’
- **StreetNamePostDirectional** - a direction after a street name, e.g. ‘North’, ‘S’
- **StreetNamePostModifier** - a modifier adter a street name, e.g. ‘Ext’
- **StreetNamePostType** - a street type that comes after a street name, e.g. ‘Avenue’, ‘Rd’

- **SubaddressIdentifier** - the name/identifier of a subaddress component
- **SubaddressType** - a level of detail in an address that is not an occupancy within a building, e.g. 'Building', 'Tower'
- **USPSBoxGroupID** - the identifier of a USPS box group, usually a number
- **USPSBoxGroupType** - a name for a group of USPS boxes, e.g. 'RR'
- **USPSBoxID** - the identifier of a USPS box, usually a number
- **USPSBoxType** - a USPS box, e.g. 'P.O. Box'
- **ZipCode** - zip code

CHAPTER 4

Important links

- Documentation: https://usaddress.readthedocs.io/
- Repository: https://github.com/datamade/usaddress
- Issues: https://github.com/datamade/usaddress/issues
- Distribution: https://pypi.python.org/pypi/usaddress
- Blog Post: http://datamade.us/blog/parsing-addresses-with-usaddress/
- Web Interface: http://parserator.datamade.us/usaddress

CHAPTER 5

Indices and tables

- genindex
- modindex
- search