OpenStreetMap Data Case Study

Map Area

Tri Cities, Washington, United States

- https://www.openstreetmap.org/export#map=11/46.2426/-119.0142

This map is of a metropolitan area to the north of my hometown, as my hometown was under 50 mb.

Problems Encountered in the Map

After downloading the map and analyzing it, the main issue that I came to find was that of street types having inconsistent names, such as ave vs avenue.

Here is some of the code I used, below:

```
#This is used to replace bad street types

mapping = { "Ave": "Avenue",

            "Pl": "Place",

            "St": "Street",

            "St.": "Street",

            "Steet": "Street",

            "ave": "Avenue",

            "Ct": "Court",

            "Dr": "Drive",

            "Blvd": "Boulevard",

            "ST": "Street",

            "Dri": "Drive"     }
```

```
#Defining the re expressions:

street_type_re = re.compile(r'\S+\.?$', re.IGNORECASE)

street_name_re = re.compile(r'.*?(?=[\wäöüß]+$)', re.IGNORECASE)

#Defining the mapping dictionary:

mapping = {"Ave": "Avenue",

            "Pl": "Place",

            "St": "Street",

            "St.": "Street",

            "Steet": "Street",

            "ave": "Avenue",

            "Ct": "Court",

            "Dr": "Drive",

            "Blvd": "Boulevard",

            "ST": "Street",

            "Dri": "Drive"

            }

#Checks to see if the element is a street name

def is_street_name(elem):

    return (elem.attrib['k'] == "addr:street")


#This function will update the street name if necessary to the correct one.

def update_name(name, mapping):

    m = street_type_re.search(name)
    o = street_name_re.search(name)
    if m:
        street_type = m.group()
        if street_type in mapping.keys():
            name = mapping[street_type]
            if o:
                firstname = o.group()


    return firstname + " " + name
#This is our first function, which is going to create and print out a dictionary of street names according to street type. It will return the dictionary.
def part1():
    st_types = audit(OSMFILE)
    pprint.pprint(dict(st_types))
    return st_types
#This is our second main function, which is going to discover the street names that are incorrect and then suggest new names for them.
def part2(street_types):
    for street_type, names in street_types.items():
            m = street_type_re.search(street_type)
            if m:
                street_type = m.group()
                if street_type in mapping.keys():

                    for name1 in names:

                        print (street_type)

                        better_name = update_name(name1, mapping)

                        print (name1, "=>", better_name)



#This is our main function here

streets = part1()

part2(streets)
```

Here are the street names before correction (these have been taken from the original TriCities.xml file that has not been cropped):

```
{'Avenue': 147, 'Street': 314, 'Drive': 102, 'Ave': 3, 'Boulevard': 54, 'W
ay': 85, 'St.': 1, 'Ct': 2, 'Court': 21, '397': 1, '44': 1, 'Road': 35, 'S
t': 3, 'Landing': 1, 'Loop': 25, '92': 1, 'Place': 6, '3920': 1, 'Parkway'
: 1, '68': 4, 'Dr': 3, 'Lane': 22, '36': 3, 'Blvd': 3, 'Trail': 3, 'ST': 1
, '72': 1, 'Dri': 1, '240': 1}
```

Here are the corrected street names:

```
Ave
West Kennewick Ave => West Kennewick Avenue
Ave
Willamette Ave => Willamette Avenue
St.
So. Kent St. => So. Kent Street
Ct
Travis Ct => Travis Court
St
```

```
W Court St => W Court Street
St
South Washington St => South Washington Street
Dr
Indian Ridge Dr => Indian Ridge Drive
Dr
Tamarisk Dr => Tamarisk Drive
Blvd
Cottonwood Creek Blvd => Cottonwood Creek Boulevard
Blvd
N Columbia Center Blvd => N Columbia Center Boulevard
ST
S OCTAVE ST => S OCTAVE Street
Dri
Clover Island Dri => Clover Island Drive
```

Data Overview and Additional Ideas

This section contains basic statistics about the dataset, the MongoDB queries used to gather them, and some additional ideas about the data in context.

File sizes

```
TriCities.xml: 84.7 mb
TriCities.xml.json: 90.1 mb
```

Number of nodes

```
#Get a count of distinct nodes
len(db.StreetData2.distinct("id", {"type" : "node"}))
```

```
1176
```

Number of ways

```
#Get a count of distinct ways
len(db.StreetData2.distinct("id", {"type" : "way"}))
```

```
47753
```

Number of unique users

```
#Get the number of distinct users
len(db.StreetData2.distinct("created.user"))
```

```
468
```

Types of Amenities

```
#What kinds of amenities are there?
db.StreetData2.distinct("amenity", {"type": "node"})
```

```
['Columbia Park East Boat Ramp',
 'atm',
 'bank',
 'bar',
 'bbq',
 'bell',
 'bench',
 'bicycle_parking',
 'bureau_de_change',
 'bus_station',
 'cafe',
 'car_rental',
 'car_sharing',
 'car_wash',
 'charging_station',
 'childcare',
 'clinic',
 'clock',
 'compressed_air',
 'dentist',
 'dojo',
 'drinking_water',
 'fast_food',
 'fire_station',
 'food_court',
 'fountain',
 'fuel',
 'ice_cream',
 'kindergarten',
 'letter_box',
 'library',
 'loading_dock',
 'marketplace',
 'music_school',
 'parking',
 'parking_entrance',
 'parking_space',
 'pharmacy',
 'place_of_worship',
 'police',
 'polling_station',
 'post_box',
 'post_office',
 'prison',
 'pub',
 'public_bookcase',
 'recycling',
 'register_office',
 'restaurant',
 'school',
 'shelter',
 'shower',
 'smoking_area',
 'social_facility',
 'telephone',
 'theatre',
 'toilets',
 'university',
 'vacuum_cleaner',
 'vending_machine',
 'veterinary',
 'waste_basket',
 'waste_disposal',
 'water_point']
```

In [105]:

List of Distinct Banks

```
#What are the names of different banks in town?
db.StreetData2.distinct("name", {"amenity": "bank"})
```

```
['Bank of America',
 'Bank of Eastern Washington',
 'Bank of the West',
 'Banner Bank',
 'Chase',
 'Columbia Bank',
 'Community First Bank',
 'First Bank',
 'GESA Credit Union',
 'Gesa Credit Union',
 'HAPO Community Credit Union',
 'Hapo Community Credit Union',
 'Hapo Credit Union',
 'KeyBank',
 'Numerica Credit Union',
 'Spokane Teachers Credit Union',
 'Sterling Savings Bank',
 'U.S. Bank',
 'Wells Fargo',
 'Yakima Federal Savings and Loan']
```

Additional Ideas

```
###The Proposal

I saw some issues in the data, where there were different types of data such as nodes and ways. It seems as if there were also the data types “public” “restriction” and “roundabout.” Is this how things are supposed to be? Further exploration is necessary. I don’t think that roundabout, restriction and public should be on the same level as node or way. I think that all three of these could be a type of node instead.

### Benefits

The benefits of implementing this suggestion would be that it would include the roundabouts, public, and restriction nodes in the .json file. Right now, those types of data are not being dealt with but ignored.

### Anticipated Problems

If there was originally a reason for roundabouts, public, and restriction to be
at the level of way or node, then it could cause unforeseen issues making them a
type of node.
```