

OpenStreetMap Case Study

Fort Worth, Texas

I enjoy visiting Fort Worth whenever I get the chance. It has several places to walk around such as the Stockyards that gives a glimpse of history on the role Fort Worth had in the cattle drives long ago.

Encountered Problems with the Fort Worth Map:

While auditing the file, I got to observe four mistakes on the abbreviations of the street names and a listing for cuisine that is not actually a cuisine.

Auditing Street Names:

```
Ave: 2
Avenue: 14990
B: 38
Blvd: 1      Dr: 1      St: 1
Boulevard: 570  Drive: 1778  Street: 9181
```

Correcting Street Names:

```
{'Ave': {'221 W. Lancaster Ave', 'Ave'},
 'Blvd': {'W Vickery Blvd'},
 'Dr': {'Juniper Dr'},
 'St': {'Main St'}}
Main St => Main Street
221 W. Lancaster Ave => 221 W. Lancaster Avenue
Ave => Avenue
Juniper Dr => Juniper Drive
W Vickery Blvd => W Vickery Boulevard
```

```
2 street_type_re = re.compile(r'\b\S+\.?$', re.IGNORECASE)
3
4 #Words included that did not need corrections
5 expected = ["Street", "Avenue", "Boulevard", "Drive", "Court", "Place", "Square", "Lane", "Road",
6             "Trail", "Parkway", "Commons", "Terrace", "West", "Way", "South", "Run", "Plaza", "North",
7             "Levee", "Highway", "Gipson", "Freeway", "East", "Drive", "Circle", "A", "B", "C", "D", "E", "G",
8             "H", "I", "J", "K", "L", "M", "N", "O", "P", "201"]
9
10 # Abbreviations that need to be fixed
11 mapping = { "St": "Street",
12             "Ave": "Avenue",
13             "Blvd": "Boulevard",
14             "Dr": "Drive"
15             }
16
17 def audit_street_type(street_types, street_name):
18     m = street_type_re.search(street_name)
19     if m:
20         street_type = m.group()
21         if street_type not in expected:
22             street_types[street_type].add(street_name)
23
```

```
def update_name(name, mapping):
    m = street_type_re.search(name)
    if m:
        street_type = m.group()
        better_type = street_type
        for problem_type in mapping:
            if street_type == problem_type:
                better_type = mapping[problem_type]

        better_name = name.replace(street_type, better_type)
    return better_name
```

Auditing Cuisine:

26

```
american: 3
barbecue: 4
burger: 6
chicken: 1
coffee_shop: 1
italian: 1
mexican: 1
mixed: 1
sandwich: 4
steak_house: 2
```

Cuisine Corrections:

Coffee_shop would need to be deleted as it is not a type of cuisine.

```
{'coffee_shop': {'coffee_shop'}}
coffee_shop =>
```








```
1 #Words included that did not need corrections
2 expected = ["american", "barbecue", "burger", "chicken", "italian", "mexican", "mixed", "sandwich", "steak_house"]
3 # Abbreviations that need to be fixed
4 mapping = { "coffee_shop": " " }
5
6
7
8
9 def audit_cuisine_type(cuisine_types, cuisine_name):
10     n = cuisine_type_re.search(cuisine_name)
11     if n:
12         cuisine_type = n.group()
13         if cuisine_type not in expected:
14             cuisine_types[cuisine_type].add(cuisine_name)
15
16
17 def is_cuisine_name(elem):
18     return (elem.attrib['k'] == "cuisine")
19
```

```
# updating the abbreviations
def update_cuisine_name(cuisine_name, mapping):
    n = cuisine_type_re.search(cuisine_name)
    if n:
        cuisine_type = n.group()
        better_cuisine_type = cuisine_type
        for problem_cuisine_type in mapping:
            if cuisine_type == problem_cuisine_type:
                better_cuisine_type = mapping[problem_cuisine_type]

        better_cuisine_name = cuisine_name.replace(cuisine_type, better_cuisine_type)
    return better_cuisine_name
```

SQL

File Size:

	nodes	7/26/2017 10:17 PM	Microsoft Excel C...	28,670 KB
	nodes_tags	7/26/2017 10:17 PM	Microsoft Excel C...	133 KB
	ways	7/26/2017 10:17 PM	Microsoft Excel C...	2,476 KB
	ways_nodes	7/26/2017 10:17 PM	Microsoft Excel C...	8,117 KB
	ways_tags	7/26/2017 10:17 PM	Microsoft Excel C...	4,479 KB
	schema.py	7/26/2017 4:27 PM	PY File	3 KB
	FTWORTH	7/25/2017 7:48 PM	XML Document	70,303 KB

The Number of Unique Users:

1	SELECT COUNT(DISTINCT(e.uid)) as 'Unique Users'
2	FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;
<	
Unique Users	
1	185

The Number of Nodes and Ways were:

1	SELECT COUNT(*) as 'Number of Nodes'
2	FROM nodes;
<	
Number of Nodes	
1	307175

1	SELECT COUNT(*) as 'Number of Ways'
2	FROM ways;
<	
Number of Ways	
1	35600

The top four listings for amenities are shown below:

1	SELECT tags.value, COUNT(*) as count
2	FROM (SELECT * FROM nodes_tags
3	UNION ALL
4	SELECT * FROM ways_tags) tags
5	WHERE tags.key='amenity'
6	GROUP BY tags.value
7	ORDER BY count desc
8	Limit 4

	value	count
1	parking	190
2	place_of_worship	168
3	restaurant	122
4	school	69

Additional Improvements:

The data quality still needs to improve. One such example is when looking at the different categories for cuisine, only 24 restaurants appear but 122 restaurants are listed above. As a result, there are 98 restaurants that do not belong to any category. Further improvements would be needed to correct this. Choosing the best category may become difficult as there are certain restaurants that could potentially fall into different categories. Additional categories would also be required to encompass unique ethnic cuisines or vegetarian restaurants that are found within the region. This would take time to research the adequate category but allow future users of Open Street Maps to browse through categories faster.

The values for parking spots also comes out with some variations. The image above shows 190 parking lots accounted for but only 134 lots are brought back below as multistory parking. There's no mention of the remaining amount. Further exploration would be required to find how the remaining lots are categorized and fix this so these lots come back with the initial lots.

26	
american:	3
barbecue:	4
burger:	6
chicken:	1
coffee_shop:	1
italian:	1
mexican:	1
mixed:	1
sandwich:	4
steak_house:	2

1	SELECT tags.value, COUNT(*) as count
2	FROM (SELECT * FROM nodes_tags
3	UNION ALL
4	SELECT * FROM ways_tags) tags
5	WHERE tags.key='parking'
6	
<	

	value	count
1	multi-storey	134

Conclusion:

Many additional things need to be done to improve the data quality of this portion of the Fort Worth map. It was truly amazing going through and identifying how much data is not in a position where it is supposed to be.