Data Wrangling with MongoDB (Final Project)

Author: Thomas Wirtz

OpenStreetMap Area: Minneapolis/St. Paul, MN United States

1. Problems Encountered in my Map

Auditing individual fields in the dataset (cities, streets, postcodes, phones, amenities, cuisines, and religious denominations) uncovered the following general issues:

- Obvious typos
- Extraneous blanks and other invalid characters
- Inconsistencies with capitalizations (e.g., 'Elk River' and 'elk river')
- Inconsistencies with abbreviations (e.g., 'Saint Paul', 'St. Paul', and 'St Paul')
- Inconsistencies with formatting (e.g., parentheses and separators in phone numbers)
- Misplaced data (e.g., cities with appended state names, postcodes containing house numbers, streets containing full addresses)
- Inconsistencies with street types: (e.g., Street, St., St)
- Missing street types (e.g., 'Arkwright' instead of 'Arkwright St')
- Inconsistencies with ordinals in streets (e.g., 'First' and '1st')
- Inconsistencies with compass point identifiers in streets (e.g., 'Northeast', 'N.E.', 'N. E.', and 'NE') and their placement at the beginning or end of the string
- Missing digits in phone numbers
- Vanity names in phone numbers (e.g., '952-746-SNAP')
- Multiple phone numbers in phone fields

My project code's addressed each of these items in its cleanup logic. To standardize street values, for example, I employed logic to:

- perform general string cleanup and fix capitalization issues
- update street types from a canonical set (abbreviations without periods): Ave, Blvd, etc.
- update compass points from a canonical set: N, S, E, W, NE, NW, SE, and SW
- move compass points from the beginning to the end of strings
- update ordinal names to their shortened forms (e.g., change First to 1st)
- update all forms of Saint to St (abbreviated without the period)

Having attended University of Minnesota, my accumulated "domain knowledge" of the Minneapolis and St. Paul areas help in resolving a few of the observed inconsistencies in the data (e.g., changing "Inver Grove" to "Inver Grove Heights"). Checking Google Maps, visiting websites of local businesses, and (in a couple of cases) reading street signs on Google Street View all helped to resolve other identified inconsistencies.

I also discovered (unsurprisingly) a fair amount of incomplete data (e.g., nodes with missing addresses, addresses without cities, and so on). While I did not attempt to fix this issue, I suggest a possible solution in Section 3.

Finally, I discovered (but did not fix) some faulty ways in the dataset:

- 187 ways with just a single node reference
- 1456 open ways with duplicate node references (i.e., first and last node references were not equal, but two other node references were the same)

2. Overview of the Data

This section provides basic statistics about the dataset and the MongoDB queries used to gather them.

File Sizes

Data File	Description	Uncompressed Size (bytes)
minneapolis-saint-paul_minnesota.osm	OSM metro extract provided by Mapzen (downloaded on May 12th, 2015)	652858628
data.json	Output file of data.py (pretty=True); input file for mongoimport command	955497356

How many total documents?

```
db.nodes.find().count()

3328978
```

How many nodes and ways?

```
db.nodes.aggregate(
      { "$group" : { "_id" : "$osm_type",
                     "count" : { "$sum" : 1 } } },
      { "$sort" : { "count" : -1 } }
).pretty()
```

```
{ "_id" : "node", "count" : 3019782 }
{ "_id" : "way", "count" : 309196 }
```

How many unique users?

```
db.nodes.aggregate(
  [
     $match :
     {
       "created.uid" : { $exists : 1 }
     }
  },
  {
     $group :
     {
       _id : "unique users",
       unique_uids : { $addToSet : "$created.uid" }
     }
  },
  {
     $unwind : "$unique_uids"
  },
  {
     $group :
     {
       _id : "$_id",
       count : { $sum : 1 }
     }
  }
).pretty()

{ "_id" : "unique users", "count" : 1197 }
```

How many prisons?

```
db.nodes.find(
  [
     amenity : "prison"
  }
).count()
```

3. Other Ideas About the Datasets

Reverse Geocoding

Each of the nodes in the dataset contained latitude and longitude values, but many of those nodes were missing address information, either in part or in total. Reverse geocoding could be used to populate missing address information in the nodes or to validate the existing address data. In this application, care would still be needed to assess the data quality of the reverse geocoder provider's results and to perform data wrangling as needed.

Additional data exploration using MongoDB queries

Find Vincent Hall (Math building at University of Minnesota):

```
db.nodes.find(
  {
      "name:en" : "Vincent Hall",
      "address.city" : "Minneapolis"
  },
  {
      _id : 0,
      osm_type : 1,
      "name:en" : 1,
      description : 1,
      operator : 1,
      "umn:BuildingCenterXYLongitude" : 1,
      "umn:BuildingCenterXYLatitude" :1
  }
).pretty()

{
      "name:en" : "Vincent Hall",
      "description" : "Department of Mathematics",
      "osm_type" : "way",
      "umn:BuildingCenterXYLatitude" : "44.97451",
      "operator" : "University of Minnesota",
      "umn:BuildingCenterXYLongitude" : "-93.234684"
}
```

Find the bar with an address nearest the Math building:

```
db.nodes.findOne(
  {
     amenity : "bar",
     address : { $exists : 1 },
     pos:
     {
       $near :
       {
         $geometry: { type: "Point",  coordinates: [ -93.234684,
44.97451 ] },
         $minDistance: 0,
         $maxDistance: 1500
       }
     }
  },
  {
     _id : 0,
     name : 1,
     address : 1,
     pos : 1
  }
)

{
     "name" : "Grumpy's Bar & Grill",
     "pos" : [
                -93.2528319,
                44.9750485
     ],
     "address" : {
                "city" : "Minneapolis",
                "street" : "Washington Ave S",
                "housenumber" : "1111",
                "postcode" : "55415"
     }
}
```

Which city has the most restaurants serving Chinese food?

```
db.nodes.aggregate(
  {
     $match :
     {
       amenity : { $in : [ "restaurant", "fast_food" ] },
       cuisine : { $regex : "chinese" },
       "address.city" : { $exists : 1 }
     }
  },
  {
     $group :
     {
       _id : "$address.city",
       count : { $sum : 1 }
     }
  },
  {
     $sort : { count : -1 }
  },
  {
     "$limit" : 1
  }
).pretty()

{ "_id" : "Cottage Grove", "count" : 3 }
```

Which two cities have the most restaurants serving the same type of food, excluding American and pizza?

```
db.nodes.aggregate(
  {
     $match :
     {
       amenity : { $in : [ "restaurant", "fast_food" ] },
       cuisine : { $exists : 1, $nin : [ "american", "pizza" ] },
       "address.city" : { $exists : 1 }
     }
  },
  {
```

```
      $group :
      {
        _id : { cuisine : "$cuisine", "address.city" :
"$address.city" },
        count : { "$sum" : 1 }
      }
  },
  {
      $sort : { "count" : -1 }
  },
  {
      "$limit" : 2
  }
).pretty()

{
      "_id" : {
                "cuisine" : "mexican",
                "address.city" : "Minneapolis"
      },
      "count" : 9
}
{
      "_id" : {
                "cuisine" : "sandwich",
                "address.city" : "St Paul"
      },
      "count" : 5
}
```