OpenStreetMap Project: Data Wrangling with MongoDB

Alf Maglalang

Map Area: Bangkok, Thailand

Name in Thai: กรุงเทพมหานคร in IPA [krūŋ tʰêːp mahǎː nákʰɔ̄ːn]
Map Bounds: minlat=12.661 minlon=99.569 maxlat=15.019 maxlon=101.337

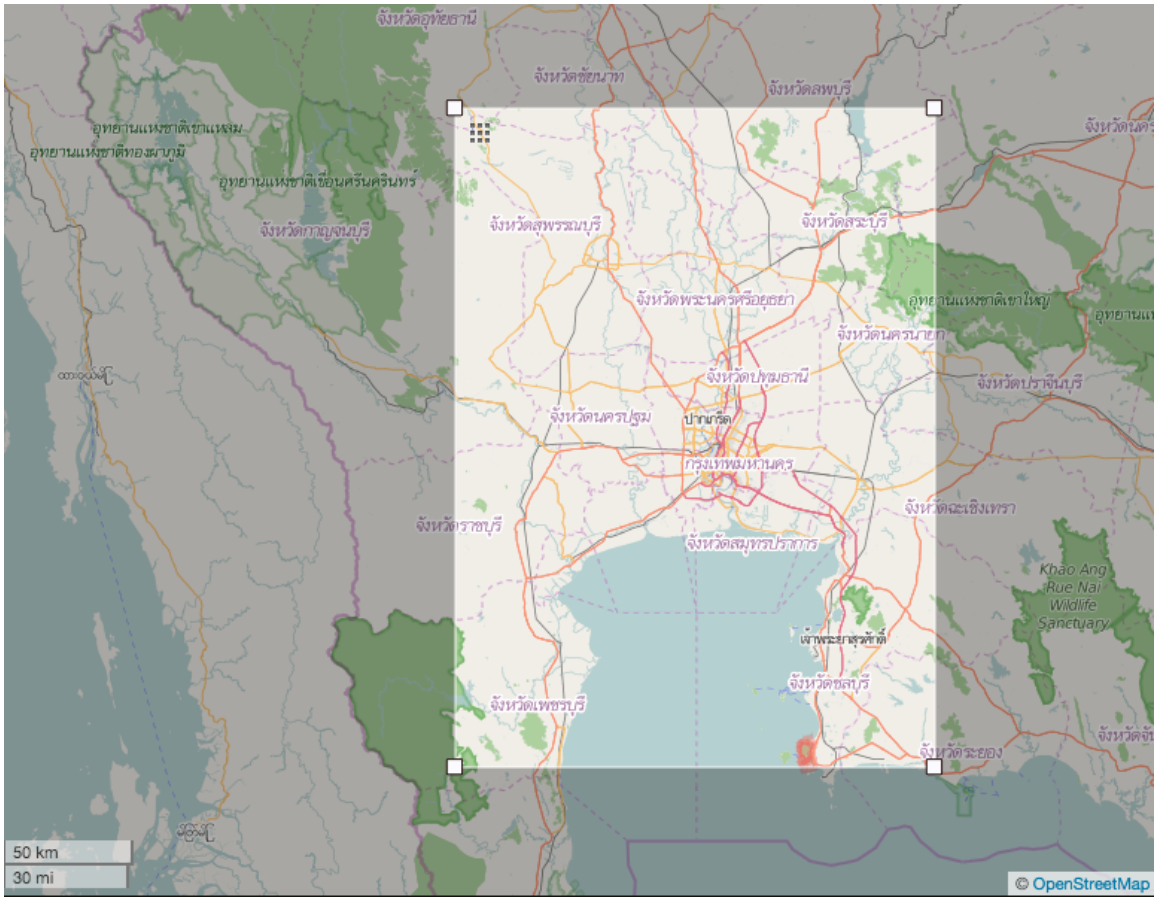
https://www.openstreetmap.org/relation/92277

1. Introduction

For this project, I have taken the opportunity to explore Bangkok with data from openstreetmap.org through MapZen's extracting site.

I have lived, studied, and worked in Bangkok since the 1980s. To me it feels like a second or third home. The chaotic 80s led to chaotic 90s into less chaotic 2000s and 2010s. These days when I go back to Bangkok, I am content in staying in my small neighborhood of Silom in Bang Rak district. But if I do venture out, I take the skytrains or the subways. What took 3 or 4 hours in the 80s to travel within the city, it now takes minutes with all the modern transportation.

As you ride the skytrains, the crowded splendor of Bangkok is in front of your eyes. There are so many streets, temples, and other places to explore. Is there some order to any of this? At least there is data to wrangle. Let's explore. The area concerned is illustrated and highlighted below.



2. Project Overview

The project is to assess the quality of the data from OpenStreetMap, in my project's case, for Bangkok, the capital and most populous city of Thailand with land area of 1900 square kilometers and population of 8,280,925.

File Sizes:

bangkok_thailand.osm: 316.5 Mb
bangkok_thailand.osm.json: 373.8 Mb
bangkok_thailand_sample.osm: 105 Kb, extract of original OSM file

Code for processing and analyzing OSM file:
gettoptags.py
validatekeys.py
k_attrib_analyzer.py
osm2jsonconverter.py
extractsampleosm.py (code from Udacity's instructor's notes)

Additional files included in project submission:
bangkok_thailand_sample.osm
bkk_pretty_sample.json
invalid_k_values.txt

The input file had the following top-level tags with their corresponding counts. I used the code gettoptags.py to retrieve the following data.

```
{ "node": 1485321,
  "nd": 1754729,
  "bounds": 1,
  "member": 8793,
  "tag": 452337,
  "relation": 913,
  "way": 200230,
  "osm": 1 }
```

For this project, I will only look at the “node” and “way” elements. Since there are 1485321 node tags and 452337 way tags, the JSON file should contain 1685551 documents. A sample OSM file bangkok_thailand_sample.osm is included in project submission compressed file.

3. Metadata Overview

Before converting the OSM file, I examined the key names (“k”) in the “tag” tag element by running validatekeys.py and k_attrib_analyzer.py. The k values fall in the 4 categories.

```
{'alphanum': 414796, 'alphanum_colon': 37244, 'other': 9, 'problemchars': 288}
```

The tagging guidelines from wiki.openstreetmap.org suggest that the tag and attribute elements be alpha-numeric characters using an underscore for more explanatory elements. Compound names can be used using a colon between words like “addr:street”. Of the total k values in the OSM file, 288 had problem characters. I examine the k values more closely by running k_attrib_analyzer.py which created an output of JSON file which I imported into mongoDB.

```
> db.key_names.distinct("key_name").length
977
> db.key_names.find({key_name:/^([a-z0-9]|_)*$/}).count()
412
> db.key_names.find({key_name:/^([a-zA-Z0-9]|_)*:([a-zA-Z0-9]|_)*$/}).count()
435
> db.key_names.find({key_name:{$not:/^([a-zA-Z0-9]|_)*(:([a-zA-Z0-9]|_)*)*$/}}).count()
36
```

Of the 977 k values 412 are completely alpha-numeric and underscore. 435 values have a colon. And only 36 unique k values make up the 288 k values with “problem” characters. Sample k values with problem characters are:

```
> db.key_names.find({key_name:{$not:/^([a-zA-Z0-9]|_)*(:([a-zA-Z0-9]|_)*)*$/}}).limit(4)
{ "_id" : "internet_access.source:fee", "total" : 1 }
{ "_id" : "Made by", "total" : 4 }
{ "_id" : "ประเภท", "total" : 1 }
{ "_id" : "name:th-Latn", "total" : 187 }
```

Examining the k values, I believe that only the k values with spacing and dot should be cleaned. Other than that, powerful regular expressions processors should be able to read alpha-numeric characters including multilingual Unicode characters and hyphens.

The top 10 key names are:

```
> db.key_names.find({ }, { "_id" : 0 } ).sort( { "count" : -1 } ).limit(10)
{ "count" : 146234, "key_name" : "highway" }
{ "count" : 54527, "key_name" : "source" }
{ "count" : 35431, "key_name" : "building" }
{ "count" : 33089, "key_name" : "name" }
{ "count" : 17584, "key_name" : "oneway" }
{ "count" : 14236, "key_name" : "name:en" }
{ "count" : 13736, "key_name" : "amenity" }
{ "count" : 9853, "key_name" : "name:th" }
{ "count" : 8298, "key_name" : "bridge" }
{ "count" : 7906, "key_name" : "ref" }
```

4. Data Overview

Using the code osm2jsonconverter.py, I produced a JSON file that will be imported into mongoDB. A sample document looks like the following:

```
{
        "_id" : ObjectId("56bfcb8b81a6c3363dbcb9f1"),
        "amenity" : "school",
        "name" : "โรงเรียนสมิทธิพงษ์",
        "created" : {
                "changeset" : "26567849",
                "user" : "charnchai",
                "version" : "2",
                "uid" : "454312",
                "timestamp" : "2014-11-05T10:39:37Z"
        },
        "pos" : [
                13.6621519,
                100.4429017
        ],
        "tagtype" : "node",
        "bkkaddress" : {
                "street" : "ถนนท่าข้าม",
                "housenumber" : "45/9",
                "postcode" : "10150"
        },
        "name_en" : "Smithipongse School",
        "name_th" : "โรงเรียนสมิทธิพงษ์",
        "id" : "1135640330"
}
```

Since both “type” and “address” are being used as values for “k” attribute, I had to use “tagtype” and “bkkaddress” as fields.

Here are some numbers about the dataset imported into mongoDB.

Total number of documents

```
> db.bkk2.find().count()
1685551
```

Number of nodes
```
> db.bkk2.find({"tagtype":"node"}).count()
1485321
```

Number of ways
```
> db.bkk2.find({"tagtype":"way"}).count()
200230
```

Number of contributors
```
db.bkk2.distinct( "created.user" ).length
1257
```

Top 10 contributing users

```
db.bkk2.aggregate([{$group:{"_id":"$created.user", "count":{"$sum":1}}}, {$sort:{"count":-1}},
{$limit:10}])
{ "_id" : "Russ McD", "count" : 264070 }
{ "_id" : "medecember", "count" : 182704 }
{ "_id" : "Paul_012", "count" : 170131 }
{ "_id" : "RocketMan", "count" : 80383 }
{ "_id" : "Parie", "count" : 54815 }
{ "_id" : "gezginrocker", "count" : 51913 }
{ "_id" : "Iain Turner", "count" : 51509 }
{ "_id" : "stephankn", "count" : 47916 }
{ "_id" : "westnordost", "count" : 47494 }
{ "_id" : "nhdr", "count" : 45555 }
```

Number of users who made 11 or fewer posts

```
db.bkk2.aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}}, {"$group":{"_id":"$count",
"num_users":{"$sum":1}}}, {"$sort":{"_id":1}}, {"$limit":12}])
{ "_id" : 1, "num_users" : 257 }
{ "_id" : 2, "num_users" : 80 }
{ "_id" : 3, "num_users" : 59 }
{ "_id" : 4, "num_users" : 46 }
{ "_id" : 5, "num_users" : 45 }
{ "_id" : 6, "num_users" : 35 }
{ "_id" : 7, "num_users" : 27 }
{ "_id" : 8, "num_users" : 23 }
{ "_id" : 9, "num_users" : 19 }
{ "_id" : 10, "num_users" : 20 }
{ "_id" : 11, "num_users" : 22 }
```

Among the 1257 unique users, 50% of the contributors made only .1% of the contribution. While the top 10 contributors made nearly 60% of the contributions. A form of incentivization should be implemented to encourage more users to input more quality and up-to-date data.

5. Problems with Data

Outdated data

There are a lot of outdated data. Bangkok changes really fast. I have seen the changes – not only with installation of modern transport, but also with creation of many new skyscrapers and businesses. The data should be updated rapidly too. Only 1.45% of the total 1685551 documents were entered in 2016, 21.32% in 2015. More than 55% of the data were entered in 2013 or earlier.

```
> db.bkk2.find({"created.timestamp":{"$exists":1}}).count()
1685551
> db.bkk2.find({"created.timestamp":{"$gte":"2016-01-01T00:00:00Z"}}).count()
24398
> db.bkk2.find({"created.timestamp":{"$gte":"2015-01-01T00:00:00Z","$lt":"2016-01-
01T00:00:00Z"}}).count()
```

```
359307
> db.bkk2.find({"created.timestamp":{"$lt":"2015-01-01T00:00:00Z","$gte":"2014-01-
01T00:00:00Z"}}).count()
367905
```

Year created	Count	Proportion from total
2016	24398	1.45 %
2015	359307	21.32 %
2014	367905	21.83 %
2013	280968	16.67 %
2012 or older	652973	38.74 %

Postal code

```
> db.bkk2.distinct( "bkkaddress.postcode" ).length
76
> db.bkk2.find({'bkkaddress.postcode':/.*/}).count()
818
> db.bkk2.find({'bkkaddress.postcode':/^([0-9]){5}$/}).count()
814
> db.bkk2.find({'bkkaddress.postcode':/[=\-\+/&<>;\'"\?%#$@\,\. \t\r\n]/}).count()
2
```

The city is so big that there are 76 postal code. There are only 818 documents that provide postal code. Postal code in Thailand only use 5 digits. There are only 4 data entry errors for postcode. 3 contain non-numeric character. And 1 contains 6 digits. These entries are:
"Tung Pong", "1212o", "102505", "10320 "

Top 3 postal codes

```
db.bkk2.aggregate([{$group:{"_id":"$bkkaddress.postcode", "count":{"$sum":1}}}, {$sort:{"count":-1}},
{$limit:3}])
{ "_id" : "10200", "count" : 89 }
{ "_id" : "10110", "count" : 83 }
{ "_id" : "10540", "count" : 66 }
```

Telephone numbers

```
> db.bkk2.find({'phone':/.*/}).count()
2073
> db.bkk2.find({'phone':/[\(\)\-; ]/}).count()
1816
```

SAMPLE:

```
{ "phone" : "+66 38 429450;+66 38 422301" }
{ "phone" : "+662-872-6955" }
{ "phone" : "+66 38 415304" }
{ "phone" : "+66 38 427142" }
{ "phone" : "(66) 2624-9555" }
```

```
> db.bkk2.find({'phone':/^([0-9])*$/}).count()
60
```

SAMPLE:

```
{ "phone" : "026356055" }
{ "phone" : "026403939" }
{ "phone" : "0879246161" }
{ "phone" : "1711" }
{ "phone" : "6626557474" }
```

```
> db.bkk2.find({'phone':/^\+66([0-9])*$/}).count()
197
```

SAMPLE:

```
{ "phone" : "+6626927000" }
{ "phone" : "+66895282428" }
```

```
{ "phone" : "+6638788226" }
{ "phone" : "+6626350111" }
{ "phone" : "+66867897833" }
```

2073 documents have phone entries. 1816 had spacing, hyphen, or semi-colon. 60 had all numeric characters. 197 had +66 (the country code) in the beginning.

I can use Standardization of telephone numbers according to wikipedia https://en.wikipedia.org/wiki/Telephone_numbers_in_Thailand. Except for Bangkok landlines, all numbers have a 3-digit area code (including the leading 0) followed by a 7-digit subscriber number. All Bangkok landlines have 02 area code. There are also old pre-1980 6-digit landline phone numbers.

I would begin the standardization with the country code by creating a new field “country_code” and use 66 as the value without the plus sign. To standardize, I would do the following:
(1) eliminate all non-numeric characters
(2) delete the initial 66 if any. Note: 66 cannot be used as an area code.
(2) from the end of the phone entry, extract the last 6 or 7 digits and use that as values for the phone.
(3) add a leading zero to the remaining digits which would be the area code
(4) prefix the area code to the 7-digit (or 6-digit) subscriber phone with a hyphen between area code and the subscriber number.

The phone numbers with 4 digits are special numbers reserved for services such as ambulance, medical assistance, government call center.

Street names

Streets in Thai sometimes begin with ถนน (which is usually a main thoroughfare) or ซอย (which branches from one of the main roads). I will ignore the English version of streets since there is no standardization of the English or romanized version of Thai streets either in speech or written form. Although used to help non-Thai speakers generally tourists or immigrant workers, the romanized names have the antithetical effect of confusing that audience. The messy sample of street names will be a challenging data wrangling task. I think all street names should be in Thai. The English or romanized version should be in another field. And in fact, there are other compound fields beginning with “name” and suffixed with a language code, for example, “name_en”, “name_th”, “name_jp”, “name_ru” respectively for English, Thai, Japanese, and Russian

SAMPLE Street Names (how to begin processing this):
```
{ "bkkaddress" : { "street" : "ทางเลียบถนนกาญจนาภิเษก ต.คลองหนึ่ง อ.คลองหลวง จ.ปทุมธานี " } }
{ "bkkaddress" : { "street" : "Petchkasem Road" } }
{ "bkkaddress" : { "street" : "Siam Cement Road (building 11)" } }
{ "bkkaddress" : { "street" : "Oriental Avenue" } }
{ "bkkaddress" : { "street" : "เพชรเกษม 76/1" } }
{ "bkkaddress" : { "street" : "ถนนเพชรเกษม" } }
{ "bkkaddress" : { "street" : "ถนนเพชรเกษม" } }
{ "bkkaddress" : { "street" : "เพชรเกษม 62/4" } }
{ "bkkaddress" : { "street" : "พุทธมณฑล สาย 1" } }
{ "bkkaddress" : { "street" : "Sukhumvit Soi 16" } }
{ "bkkaddress" : { "street" : "Sukhumvit Soi 16" } }
{ "bkkaddress" : { "street" : "ถนนจรัญสนิทวงศ์" } }
{ "bkkaddress" : { "street" : "New Petchburi Road" } }
{ "bkkaddress" : { "street" : "ถนนพัฒนการ" } }
{ "bkkaddress" : { "street" : "Sukhumvit Road " } }
{ "bkkaddress" : { "street" : "ถนนหลังสวน" } }
{ "bkkaddress" : { "street" : "ถนนหลังสวน" } }
{ "bkkaddress" : { "street" : "ถนนคนเดิน สวนจตุจักร โซน2" } }
{ "bkkaddress" : { "street" : "Sukhumvit" } }
{ "bkkaddress" : { "street" : "ถนนกัลปพฤกษ์" } }
```

6. Further Exploration of Dataset

Top 10 amenities. People who live in Bangkok love to eat. Street food vendors are everywhere. What the Bangkok dataset probably does not include are the thousands of street eating possibilities all over the city. Here are the top 10 amenities in Bangkok.

```
> db.bkk2.aggregate([{$match:{"amenity":{$exists:1}}},{ $group: { "_id": "$amenity", total: { $sum: 1 }
} },{$sort: {"total":-1}} ,{$limit:10}])
{ "_id" : "restaurant", "total" : 2211 }
{ "_id" : "place_of_worship", "total" : 1397 }
{ "_id" : "parking", "total" : 1224 }
{ "_id" : "fuel", "total" : 1004 }
{ "_id" : "bar", "total" : 950 }
{ "_id" : "cafe", "total" : 787 }
{ "_id" : "bank", "total" : 538 }
{ "_id" : "school", "total" : 533 }
{ "_id" : "telephone", "total" : 501 }
{ "_id" : "atm", "total" : 406 }
```

Top 10 cuisines. 203 cuisines are represented in the dataset

```
> db.bkk2.distinct( "cuisine" ).length
203
> db.bkk2.aggregate([{$match:{"cuisine":{$exists:1}}},{ $group: { "_id": "$cuisine", total: { $sum: 1 }
} },{$sort: {"total":-1}} ,{$limit:10}])
{ "_id" : "thai", "total" : 234 }
{ "_id" : "coffee_shop", "total" : 127 }
{ "_id" : "japanese", "total" : 85 }
{ "_id" : "burger", "total" : 81 }
{ "_id" : "pizza", "total" : 53 }
{ "_id" : "indian", "total" : 50 }
{ "_id" : "chicken", "total" : 48 }
{ "_id" : "italian", "total" : 46 }
{ "_id" : "regional", "total" : 41 }
{ "_id" : "international", "total" : 37 }
```

9 religions are represented in the dataset. Nearly 95% of Thais are Buddhist. This explains why the Buddhist representation in the dataset is 90%.

```
> db.bkk2.distinct( "religion" ).length
9

> db.bkk2.aggregate([{$match:{"religion":{$exists:1}}},{ $group: { "_id": "$religion", total: { $sum: 1
} } },{$sort: {"total":-1}}])
{ "_id" : "buddhist", "total" : 1191 }
{ "_id" : "christian", "total" : 53 }
{ "_id" : "muslim", "total" : 45 }
{ "_id" : "taoist", "total" : 13 }
{ "_id" : "hindu", "total" : 13 }
{ "_id" : "buddist", "total" : 2 }
{ "_id" : "sikh", "total" : 2 }
{ "_id" : "spiritualist", "total" : 1 }
{ "_id" : "jewish", "total" : 1 }
```

7. Resources

– Udacity's course on Data Wrangling with MongoDB
– https://mapzen.com/data/metro-extracts
– http://www.json.org/
– https://wiki.python.org/moin/PythonXml
– https://docs.mongodb.org/getting-started/python/
– http://wiki.openstreetmap.org/wiki/Map_Features
– https://en.wikipedia.org/wiki/Telephone_numbers_in_Thailand
– https://en.wikipedia.org/wiki/Bangkok