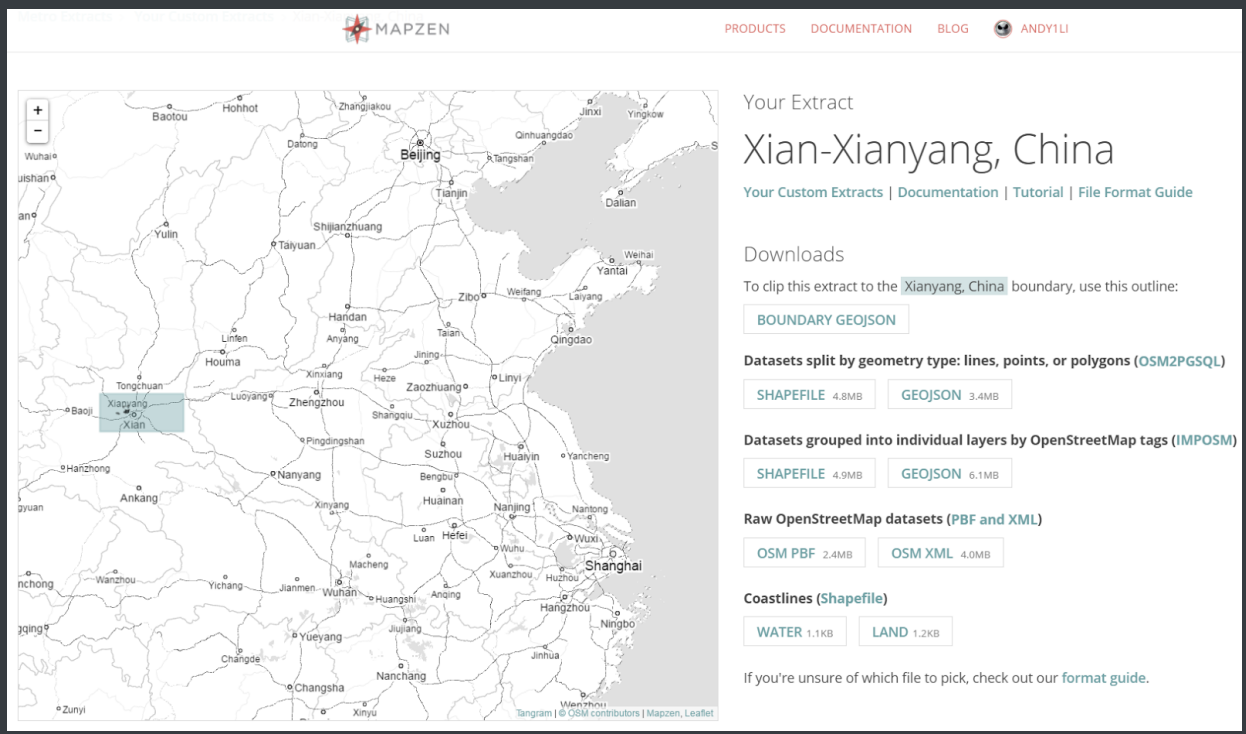
OpenStreetMap Data Wrangling

OpenStreetMap Data Wrangling

The chosen map area encompasses cities of Xi'an and Xianyang (my hometown) and tourist attractions such as Mount Hua in Shaanxi province of inner China.



Raw OSM XML:
https://s3.amazonaws.com/mapzen.odes/ex_NZpwL5ga4V4e3Yu2fFLZW25Br25gy.osm.bz2

1. Problems

After familiarizing with the structure of the data and exploring in the `audit.ipynb` file, I have identified the following major problems:

- `'name'` tags which should contain only Chinese characters actually contain:
  - only English letters (`'Grand Park Xian'`) or
  - mixed Chinese-English characters (`'兵马俑 terracotta army'`)
  - homophonic typos (`'高薪四路 / Gaoxin Si Lu'`)
- Overly-abbreviated `'name:en'` tags (`'Weiyang West Rd'` , `'Ring Expy'`)

Only English in `'name'` tags

This problem can be easily solved when there are quality `'name:zh'` tags as the gold standard (but difficult otherwise).

```
ALL_EN = re.compile(u'^[a-zA-Z0-9\\s]+$')
ALL_CN = re.compile(u'^[\u4e00-\u9fa5]+$')

if tag['key'] == 'name' and ALL_EN.match(value):
    # Use name:zh as the gold standard
    zh_tag = get_zh_tag(tags)
    if zh_tag and ALL_CN.match(zh_tag['value']):
        tags[i]['value'] = zh_tag['value']
```

For example, this approach fixed tags perfectly such as:

```
<tag k="name" v="Gaoling Xian"/>
<tag k="name:zh" v="高陵县"/>
```

Mixed Chinese-English in `'name'` tags

Of course, the `'name:zh'` gold standard still applies. However, in the absence of it, only some of the tags can be fixed easily (when Chinese can be cleanly extracted before a 'space').

```
CN_EN  = re.compile(u'^[\u4e00-\u9fa5\\s]+[^\u4e00-
\u9fa5]+$')
CN_SPACE_EN = re.compile(u'^[\u4e00-\u9fa5\\s]+\\s[^\u4e00-
\u9fa5]+$')

if tag['key'] == 'name' and CN_EN.match(value):
    # Fix tags with a space separating the Chinese
    # and English parts, such as:
    # '兵马俑 terracotta army'
    # '高薪四路 / Gaoxin Si Lu'
    if CN_SPACE_EN.match(value):
        cn, _, _ = value.partition(' ')
        tags[i]['value'] = cn
```

Partial results:

```
Before:  兵马俑 terracotta army
After:   兵马俑

Before:  高薪四路 / Gaoxin Si Lu
After:   高薪四路
```

Homophonous typos in `'name'` tags

Amusing typos surfaced during the auditing process. For example, "高薪" (high paying) had been used instead of the presumed "高新" (high tech), which sound the same, look alike, and result in non-sequitur humorous effects, such as "4th High Paying Road" and "No. 1 High Paying Elementary School". 🤣

The typo problems have a similar solution to that of the overly-abbreviated tags.

```
TYPOS = {u'高薪': u'高新'}

for wrong, right in TYPOS.items():
    if wrong in value:
        tags[i]['value'] = tags[i]['value'].replace(wrong,
right)
```

Overly-abbreviated `'name:en'` tags

I won't waste dear reviewer's time by dwelling too much on this one, since it's probably the same thing over and over again, for thousands of times already.

```
ABBRES = {
    'Blvd': 'Boulevard',
    'Rd': 'Road',
    'Lu': 'Road',
    ...,
    'Qu': 'District',
}

for abbre, full in ABBRES.items():
    if value.endswith(abbre)
        tags[i]['value'] = tags[i]['value'].replace(abbre,
full)
```

---

2. Data Overview

File Size

Xian-Xianyang.osm: 54.8 MB

Number of Elements

```
SELECT COUNT(*) FROM nodes;      -- 256,226
SELECT COUNT(*) FROM nodes_tags; --   8,159
SELECT COUNT(*) FROM ways;       --  34,149
SELECT COUNT(*) FROM ways_tags;  --  71,474
```

Top 5 Tag Types

```
SELECT type, COUNT(*)
FROM (
    SELECT type FROM nodes_tags UNION ALL
    SELECT type FROM ways_tags
)
GROUP BY type
ORDER BY COUNT(*) DESC
LIMIT 5;
```

	type	COUNT(*)
1	regular	75085
2	name	3169
3	addr	527
4	lanes	492
5	building	135

Top 5 Regular Keys

```
SELECT key, COUNT(*)
FROM (
    SELECT key FROM nodes_tags WHERE type="regular"
    UNION ALL
    SELECT key FROM ways_tags WHERE type="regular"
)
GROUP BY key
ORDER BY COUNT(*) DESC
LIMIT 5;
```

	key	COUNT(*)
1	highway	22142
2	name	8774
3	building	6478
4	oneway	6352
5	source	4035

Top 5 Sources

```
SELECT value, COUNT(*)
FROM (
    SELECT value FROM nodes_tags WHERE key="source"
    UNION ALL
    SELECT value FROM ways_tags WHERE key="source"
)
GROUP BY value
ORDER BY COUNT(*) DESC
LIMIT 5;
```

	value	COUNT(*)
1	Bing	1208
2	GPS	1161
3	bing	953

4	osm-gpx	302
5	mapbox	117

---

3. Additional Ideas

The last query of the previous section exposes a problem I overlooked: both the first and third most common sources are Bing (just in different cases), which definitely should be cleaned in the next iteration.

This naturally suggests how to further improve the data: incorporating map data as the gold standard from other large corporations, such as Google, Baidu, and so on.

The benefits are huge and obvious. Those big corporations have the latest, most comphresive map data out there. There does not seems to be any reason not to utilize them.

Or is there? The most glaring problem is to come up with a clever plan to merge the data from different sources, and reconcile any duplications and inconsistencies. On top of that, geographic data from China are randomly offseted due to "security reasons", which will unnecessarily complicate any plan we come up with.

Besides, legal concerns cannot be ignored. Do those corporations allow their data to be incorporated by others freely? We should check this issue out before doing anything, in case they do something as counterstrikes.

References

Restrictions on geographic data in China