A Turbo Introduction to Overpass

Minh Nguyễn
State of the Map U.S. 2019
<table>
<thead>
<tr>
<th>State</th>
<th>File Format</th>
<th>Size</th>
<th>Source Format</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connecticut</td>
<td>[.osm.pbf]</td>
<td>(27.4 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Delaware</td>
<td>[.osm.pbf]</td>
<td>(10.5 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>District of Columbia</td>
<td>[.osm.pbf]</td>
<td>(15.8 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Florida</td>
<td>[.osm.pbf]</td>
<td>(220 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Georgia (US State)</td>
<td>[.osm.pbf]</td>
<td>(175 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Hawaii</td>
<td>[.osm.pbf]</td>
<td>(11.9 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Idaho</td>
<td>[.osm.pbf]</td>
<td>(62 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Illinois</td>
<td>[.osm.pbf]</td>
<td>(166 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Indiana</td>
<td>[.osm.pbf]</td>
<td>(69 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Iowa</td>
<td>[.osm.pbf]</td>
<td>(84 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Kansas</td>
<td>[.osm.pbf]</td>
<td>(58 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Kentucky</td>
<td>[.osm.pbf]</td>
<td>(104 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Louisiana</td>
<td>[.osm.pbf]</td>
<td>(99 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Maine</td>
<td>[.osm.pbf]</td>
<td>(43.8 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Maryland</td>
<td>[.osm.pbf]</td>
<td>(132 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Massachusetts</td>
<td>[.osm.pbf]</td>
<td>(235 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Michigan</td>
<td>[.osm.pbf]</td>
<td>(128 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Minnesota</td>
<td>[.osm.pbf]</td>
<td>(169 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
<tr>
<td>Mississippi</td>
<td>[.osm.pbf]</td>
<td>(68 MB)</td>
<td>[.shp.zip]</td>
<td></td>
</tr>
</tbody>
</table>
Overpass API Query Form

[out:json][timeout:25];
(  way[-"turn"~".*"](user:"Minh Nguyen");
);
out body;
>;
out skel qt;

Overpass API Convert Form
/*
This has been generated by the overpass-turbo wizard.
The original search was:
"man_made=flagpole and type:node and
user:"Minh Nguyen"
*/
[out:json][timeout:25];
// gather results
{
  // query part for: "man_made=flagpole
  and user:"Minh Nguyen"
  node["man_made"="flagpole"](user:"Minh Nguyen")
  {{bbox}};
}
// print results
out body;
>
out skel qt;
overpass-turbo.eu
This has been generated by the overpass-turbo wizard.
The original search was:

"traffic_calming=bump and type:node and user:"Minh Nguyen"  global"

[out:json][timeout:50];

// gather results
(
    // query part for: "traffic_calming=bump and user:"Minh Nguyen"

    node["traffic_calming"="bump"]
    (user:"Minh Nguyen");
);

// print results
out body;
>;
out skel qt;
amenity=school
(amenity=school or amenity=university) and (type:way or type:relation)
amenity=school
and name~"High School"

Regular expressions

© OSM contributors, ODbL
Saint Paul, Minnesota
amenity=school
in "Saint Paul, Minnesota"
amenity=school and operator!=* in "Saint Paul, Minnesota"
More wizardry

- amenity=school and user:"Marvelous Mapper"
- amenity=school newer:1day
- ~(disused:|demolished:)?amenity~"school"
- amenity=language_school global
amenity=school

[out:json][timeout:25];
(
    node["amenity"="school"]
    ({{bbox}});
    way["amenity"="school"]
    ({{bbox}});
    relation["amenity"="school"]
    ({{bbox}});
);
out body;
>;
out skel qt;
amenity=school

[out:json][timeout:25];
(
  node["amenity"="school"]({{bbox}});
  way["amenity"="school"]({{bbox}});
  relation["amenity"="school"]
   ({{bbox}});
);
out body;
>;
out skel qt;
amenity=school

[out:json][timeout:25];
(
  node["amenity"="school"]({{bbox}});
  way["amenity"="school"]({{bbox}});
  relation["amenity"="school"]
    ({{bbox}});
);
out body;
>;
out skel qt;
amenity=school

[out:json][timeout:25];

( node["amenity"="school"]({{bbox}})
  way["amenity"="school"]({{bbox}})
  relation["amenity"="school"]({{bbox}})
);

out body;
>
out skel qt;
amenity=school

[out:json][timeout:25];
(
  node["amenity"="school"]({{bbox}});
  way["amenity"="school"]({{bbox}});
  relation["amenity"="school"]
    ({{bbox}});
);  
out body;  
>;
out skel qt;
amenity=school

[out:json][timeout:25];
(
  node["amenity"="school"]
  ({{bbox}});
  way["amenity"="school"]
  ({{bbox}});
  relation["amenity"="school"]
  ({{bbox}});
);
out body;
>;
out skel qt;
amenity=school and type:way

[out:json][timeout:25];
(
    way["amenity"="school"]
    ({{bbox}});
);
out body;
>;
out skel qt;
(amenity=school or amenity=university) and (type:way or type:relation)

[out:json][timeout:25];
(
  way["amenity"="school"]
  ({{bbox}});
  relation["amenity"="school"]
  ({{bbox}});
  way["amenity"="university"]
  ({{bbox}});
  relation["amenity"="university"]
  ({{bbox}});
);
out body; >; out skel qt;
amenity=school and name~"High School"

[out:json][timeout:25];
(
  node["amenity"="school"]
  ["name"~"High School"]({{bbox}});
  way["amenity"="school"]
  ["name"~"High School"]({{bbox}});
  relation["amenity"="school"]
  ["name"~"High School"]({{bbox}});
);
out body;
>;
out skel qt;
amenity=school in "Saint Paul, Minnesota"

[out:json][timeout:25];

{{{geocodeArea:Saint Paul, Minnesota}}} ->.searchArea;

( node["amenity"="school"]
  (area.searchArea);
  way["amenity"="school"]
    (area.searchArea);
  relation["amenity"="school"]
    (area.searchArea);
);

out body; >; out skel qt;
amenity=school and operator!="*"

[out:json][timeout:25];

( node["amenity"="school"]
  ["operator"!=".*"]({{bbox}}); way["amenity"="school"]
  ["operator"!=".*"]({{bbox}}); relation["amenity"="school"]
  ["operator"!=".*"]({{bbox}}));

out body; >; out skel qt;
Using the Results
Permalink
Copy this link to share the current code:
http://overpass-turbo.eu/s/M6V

Options
- include current map state
- run this query immediately after loading

Export

- download/copy as GeoJSON
- download/copy as GPX
- download/copy as KML
- download/copy as raw OSM data
- raw data directly from Overpass API
- load data into an OSM editor: JOSM, Level0
- save GeoJSON to gist

Map

Query

done
MapRoulette
Mapbox Studio

tinyurl.com/sotmus18heatmaps2
JOSM

Expert Mode
Footways intersecting roadways

[out:json][timeout:25];

way["highway"!="footway"]
  ({{bbox}});
node(w);
way(bn)["highway"="footway"];  

out body;
>
out skel qt;
Footways intersecting roadways

```json
[out:json][timeout:25];

way["highway"!="footway"] ({{bbox}});
node(w);
way(bn)["highway"="footway"];

out body;
>;
out skel qt;
```

© OSM contributors, ODbL
Footways intersecting roadways

[out:json][timeout:25];

way["highway"!="footway"] ({{bbox}});
node(w);
way(bn)["highway"="footway"];

out body;
>;
out skel qt;
Footways intersecting roadways

[out:json][timeout:25];

way["highway"!="footway"]
  ({{bbox}});
node(w);
way(bn)["highway"="footway"];
Buildings near tornado sirens

```json
[out:json][timeout:25];

node["emergency"="siren"] ({{bbox}});
way(around:100)["building"];

out body;
>
out skel qt;
```
Buildings near tornado sirens

```
[out:json][timeout:25];

node["emergency"="siren"]
  ({{bbox}});
way(around:100)["building"];

out body;
>;
out skel qt;
```
Buildings near tornado sirens

```json
[out:json][timeout:25];
node["emergency"="siren"] ({{bbox}});
way(around:100)["building"];
out body; >;
out skel qt;
```
Flagpoles not in front of fire stations

```json
[out:json][timeout:25];
node["man_made"="flagpole"]({{bbox}}) -> .flagpoles;
(
  node["amenity"="fire_station"]({{bbox}});
  way["amenity"="fire_station"]({{bbox}});
); 
node["man_made"="flagpole"](around:100) -> .firepoles;
(node.flagpoles; - node.firepoles;);
out body;
```

© OSM contributors, ODbL
Flagpoles not in front of fire stations

[out:json][timeout:25];
node["man_made"="flagpole"]({{bbox}})
  ->.flagpoles;
(
  node["amenity"="fire_station"]
  ({{bbox}});
  way["amenity"="fire_station"]
  ({{bbox}}));
);
node["man_made"="flagpole"]
  (around:100)->.firepoles;
(node.flagpoles; - node.firepoles;);
out body;
Flagpoles not in front of fire stations

[out:json][timeout:25];
node["man_made"="flagpole"]({{bbox}})
  ->.flagpoles;
(
  node["amenity"="fire_station"]
  ({{bbox}});
  way["amenity"="fire_station"]
  ({{bbox}});
);
node["man_made"="flagpole"]
  (around:100)->.firepoles;
(node.flagpoles; - node.firepoles;);
out body;
Flagpoles not in front of fire stations

```json
[out:json][timeout:25];
node["man_made"="flagpole"](bbox)
  ->.flagpoles;
(
  node["amenity"="fire_station"]
  (bbox);
  way["amenity"="fire_station"]
  (bbox);
); node["man_made"="flagpole"]
  (around:100)->.firepoles;
(node.flagpoles; - node.firepoles;);
out body;
```
Flagpoles not in front of fire stations

```json
[out:json][timeout:25];
node["man_made"="flagpole"]({{bbox}})
  ->.flagpoles;

(  
  node["amenity"="fire_station"]
    ({{bbox}});  
  way["amenity"="fire_station"]
    ({{bbox}});  
);
node["man_made"="flagpole"]
  (around:100)->.firepoles;
(node.flagpoles; - node.firepoles;);
out body;
```
Long turn lanes

```json
[out]:json[timeout:25];

way[~"^turn"~".*"]({{bbox}})
  (if: length() > 300);

out body;
>
out skel qt;
```
Total length of bike infrastructure

```
[out:json][timeout: 25];
(
    {{geocodeArea: Minneapolis}};
    {{geocodeArea: Saint Paul, Minnesota}};
)->.searchArea;

(  way[~"cycleway"~"lane|track|opposite|shared_lane|shared|crossing|opposite_lane|yes|share_busway"](area.searchArea);
  way["highway"="cycleway"](area.searchArea);
  way["bicycle"~"yes|permissive|designated"](area.searchArea);
);

make stats length=sum(length());
out;
```

490½ centerline miles
Performance

- Avoid overspecifying filters

- Increase timeout: [timeout:60] (1 minute)

- Increase memory: [maxsize:1073741824] (1 GB)

- Check the rate limiting status for your IP address: overpass-api.de/api/status
Thank You

- Special thanks to Roland Olbricht, Martin Raifer, mmd
- osmlab.github.io/learnoverpass
- wiki.openstreetmap.org/wiki/Overpass_API
- #overpass on OSMUS Slack
- minh@mapbox.com or minh@nguyen.cincinnati.oh.us
- @1ec5

© David Herrera, CC BY