Limpando dados do OpenStreetMap com MongoDB

Visão geral do projeto

Para realizar a limpeza dos dados, foi escolhida a área de Boston, EUA, no https://www.openstreetmap.org e utilizado as técnicas de tratamento para avaliar a qualidade dos dados para validade, precisão, plenitude, consistência e uniformidade. Por último, foi escolhido MongoDB como modelo para armazenamento dos dados limpos, para, enfim, realizar análise exploratória nos dados.

Área do Mapa: Boston, EUA

https://www.openstreetmap.org/export#map=12/42.3677/-71.0458

1. Problemas encontrados

Após coletar os dados do Open Street Map, resolvi realizar uma rápida auditoria nos dados, especificamente nos dados de endereço, e foi detectado dois problemas que precisam ser resolvidos.

- Diversas abreviações para o mesmo endereço (Street = St, st, ST, St, St)
- Códigos postais inválidos ('MA', 'MA 02116', 'MA 02135', 'MA')

Diversas formas de abreviações.

Para resolver os problemas de abreviações, foi necessário definir uma lista com os valores desejados, a fim de realizar a substituição dos valores problemáticos.

Exemplo: Somerville Ave => Somerville Avenue
Massachusetts Ave => Massachusetts Avenue
Main St. => Main Street

Códigos postais inválidos.

Alguns códigos postais tinham letras (MA 02116) em sua composição e estavam fora do padrão utilizado nos EUA (12345-6789) ou (12345). Para resolver a situação, fui obrigado a remover as letras dos códigos que tinham letras em sua composição e os que não tinha números foram totalmente ignorados.

2. Importando os dados no Banco de Dados MongoDB

Antes de importar os dados no MongoDB, precisamos realizar uma limpeza nos problemas encontrados na auditoria e estruturar os dados de uma forma que facilite sua utilização posteriormente.

4. Visão Geral dos Dados

Esta seção demonstrará algumas estatísticas básicas sobre os dados, após importá-los no MongoDB.

Tamanho dos arquivos

boston.osm..... 76MB

boston.osm.json..... 83.5MB

Número de documentos

db.streetmap.find().count()

- Existem 779976 documentos cadastrados

Número de nós

db.streetmap.find({"type" : "node"}).count())

- Existem 335305 nodes cadastrados

Número de caminhos

db.streetmap.find({"type" : "way"}).count())

- Existem 54591 ways cadastrados

Número de usuários únicos

db.streetmap.distinct("created.user")

- Existem 906 usuários cadastrados

Número de dormitórios

db.streetmap.find({'building.building': "dormitory"}

- *Existem 49 dormitórios cadastrados*

Número de universidades

db.streetmap.find({'building.building': "university"}

- Existem 175 universidades cadastradas

Lista dos 10 usuários que mais contribuíram

```
group = {"$group":{"_id" : "$created.user", "count": {"$sum": 1}}}
sort  = {"$sort":{"count": -1}}
limit = {"$limit": 10}
pipeline = [group, sort, limit]
db.streetmap.aggregate(pipeline)
```

USUÁRIO	NÚMERO DE CONTRIBUIÇÕES
crschmidt	182410
jremillard-massgis	40533
wambag	26930
morganwahl	23147
ryebread	18951
OceanVortex	13009
mapper999	9411
cspanring	5687
JasonWoof	4651
synack	4211

Lista das 10 origens que mais contribuíram

```
group = {"$group":{"_id":"$source", "count":{"$sum": 1}}}
sort  = {"$sort":{"count": -1}}
limit = {"$limit":10}
pipeline = [group, sort, limit]
db.streetmap.aggregate(pipeline)
```

ORIGEM DOS DADOS	NÚMERO DE CONTRIBUIÇÕES
Desconhecida	376919
massgis_import_v0.1_20071008193615	5595
massgis_import_v0.1_20071008165629	2282
massdot_import_081211	844
massgis_import_v0.1_20071009093301	769
Bing	748
USGS Geonames	333
massgis_import_v0.1_20071013192438	284
massgis_import_v0.1_20071009094247	282
massgis_import_v0.1_20071008141127	262

5. Reflexão sobre os dados

Ao analisar os dados, me deparei com alguns problemas na estruturação e nos valores encontrados para alguns campos. O problema que chamou mais a atenção, foi o fato de que em alguns elementos existir a tag "address" com o agrupamento das informações de endereço e sem padrão algum. Eu recomendaria incentivar a não utilização dessa tag e estruturar os dados de endereço somente na tag "addr" como vimos em algumas entradas. Outra recomendação seria utilizar um xml com definição de tipos, a fim de evitar problemas com dados sujos. Pensando nas contribuições dos registros, podemos observar que 10 usuários contribuíram com cerca de 50% dos dados, como nem todas as pessoas tem conhecimento técnico para ficar enviando dados para atualizar os registros, eu recomendaria buscar parceria com redes sociais e outros aplicativos para atualizar os dados por meio de marcação das fotos nos locais ou recomendações.

1. Agrupamento das informações de endereço na tag "address".
    - **Solução**: incentivar a não utilização dessa tag e estruturar os dados de endereço somente na tag "addr" como vimos em algumas entradas.
        - **Benefícios:**
            - Facilidade na extração das informações de endereço nas tags estruturados.
        - **Problemas esperados:**
            - Revisão nos programas que realizavam extração das informações da tag "address".
2. Tipos de dados incompatíveis. Ex: código postal: 'MA 02116'
    - **Solução:** utilizar um xml com definição de tipos.
        - **Benefícios:**
            - Dados mais padronizados e fáceis de processar
            - Evitará problemas com dados sujos.
        - **Problemas esperados:**
            - Diminuição da flexibilidade de implementação dos programas.
            - Aumentará o tempo para o desenvolvimento das soluções.
3. Poucos usuários contribuem com a atualização dos dados:
    - **Solução:** buscar parcerias com redes sociais para a inclusão de dados no OSM.
        - **Benefícios:**
            - Aumento direto nas contribuições dos dados, devido ao grande número de usuários nas redes sociais.
            - Dados mais confiáveis.
        - **Problemas esperados:**
            - Melhoria na infraestrutura dos servidores, devido a grande quantidade de requisições com origem nas redes sociais.

6. Conclusão

Neste projeto, escolhi analisar a área de Boston, EUA, e foi realizado o processo de limpeza de dados. Na fase de auditoria, encontrei alguns problemas como códigos postais inválidos e diversas abreviações diferentes para o mesmo endereço. Em seguida, realizei as devidas correções e importei os dados limpos no MongoDB. Para evitar o imenso esforço para: auditar, limpar e resubmeter os dados, deveria padronizar a entrada de dados por Bots inteligentes ou por órgãos como o MassGIS - Escritório de Informação

Geográfica e Ambiental da Commonwealth, a fim de garantir dados mais limpos que os inseridos manualmente.