Projeto: Limpando dados do OpenStreetMap

- Aluno: Eric Inohira Uchimura
- email: euchimura@gmail.com

Visão Geral do Projeto

Você escolherá qualquer área do mundo em https://www.openstreetmap.org (https://www.openstreetmap.org) e utilizará técnicas de tratamento, como avaliar a qualidade dos dados para validade, precisão, plenitude, consistência e uniformidade, limpando os dados do OpenStreetMap de uma parte do mundo que você se importa. Por último, você vai escolher entre MongoDB e SQL como modelo de dados para completar o seu projeto.

Dados Escolhidos - DataSet

O Mapa selecionado foi o Mapa da cidade de Vitória do Estado do Espirito Santo, Brasil. Esta área foi selecionada por questões emocionais do aluno, primeiro por ser nativo dessa região e em segundo plano, por conhecer as ruas, avenidas e locais ali presentes.

http://www.openstreetmap.org/export#map=13/-20.2848/-40.2798
(http://www.openstreetmap.org/export#map=13/-20.2848/-40.2798)

Os dados foram extraidos utilizando a API do OverPass (http://overpass-api.de/query_form.html (http://overpass-api.de/query_form.html)) no dia 21/09/2017 utilizando o seguinte comando

- (node(-20.3264, -40.3602, -20.2399,-40.2081);<;);out meta;

Tamanho do arquivo: 167.428 kb

Limpeza dos Dados - Data Audit

Inicialmente criou-se um procedimento "auditoria" que percorre o arquivo OSM e verifica os nomes de ruas que não estão na lista de nome de rua Esperados. Notou-se os seguintes problemas nos dados:

- Os caracteres contendo acento e cedilha apareceram com codigo \uxxxxx
- Abreviação das ruas utilizando o R. - Neste caso, nao poderia substituir todas as ocorrencias de "R." por "Rua", uma vez que existia nomes de rua com "R.". Como Exemplo temos o beco chamado "Gumercindo R. de Souza". Neste caso o nome do Beco não seria "Gumercindo Rua de Souza".
- Abreviação de avenidas utilizando "Av." ou "Av ".
- Abreviação de alamedas utilizando "Al."
- Abreviação de rodovias utilizando "Rod.",
- Abreviação de edificios utiliando "Ed.",
- pt: que neste caso é ponto de referencia. Neste caso os valores que começarem com "pt" é gravado no dicionario de dados como "pontoReferencia"

Para solucionar os problemas de acento e cedilha foi gerado uma função auxiliar, que recebe como entrada uma string e retorna uma string sem os acentos e cedilhas

In [2]:

```
def remover_acentos(txt, codif='utf-8'):
    txt = txt.encode('utf-8')
    return normalize('NFKD', txt.decode(codif)).encode('ASCII','ignore')
```

Para os problemas de abreviações, desenvolveu-se uma função "auditaNome" que recebe 3 variáveis de entrada:

- String contendo o texto
- Dicionário para substituição no inicio (mapInicio) - Para tratar o problema do "R."
- Dicionário para subsituição de abreviações em geral que acontecem tanto no inicio quanto no meio da string (mapFim)

Foram utilizados as funções "shapeElement" e "process_map" - baseados nos exercícios - com pequenas modificações para capturar as tags e os seus respectivos conteúdos e salvar em um arquivo ".json"

Nota: para dados do openstreetMap com tamanho de 1GB ou mais, a função "process_map" falha após 1,2 milhão de interações por falta de memória. Ao utilizar o Python 3, as interações aumentam para 3,3 milhões, porém também dá erro de falta de memória. Para isso foi criado uma outra função, que ao invés de gravar diretamente em um arquivo texto, o programa grava cada interação diretamente no Banco de dados.

Importação dos dados para o MongoDB

Após rodar a função "process_map", foi gerado um arquivo Json contendo as informações limpadas e organizadas para posterior importação no banco.

- Arquivo: Vitoria.txt.json, gerado no dia 27/09/2017 com tamanho de 168.445kb

Utilizando o terminal do CommandPrompt foi importado o arquivo JSON para o banco de dados utilizando o seguinte comando

- mongoimport -db udacity --colletion osmdata --drop --file Vitoria.txt.json

Informações do Banco de Dados Importado

Por meio do comando db.stats() no shell do MongoDb temos as seguintes informações:

In [ ]:

```
> db.stats()
{
        "db" : "udacity",
        "collections" : 1,
        "views" : 0,
        "objects" : 786231,
        "avgObjSize" : 231.15248953551819,
        "dataSize" : 181739253,
        "storageSize" : 51339264,
        "numExtents" : 0,
        "indexes" : 1,
        "indexSize" : 7643136,
        "ok" : 1
}
```

Quantidade de documentos:

- 786231

In [ ]:

```
 db.getCollection('osmdata').find().count()
786231
```

Quantidade de Nodes

- 695903

In [ ]:

```
db.getCollection('osmdata').find({"type":"node"}).count()
695903
```

Quantidade de Ways

- 90327

In [ ]:

```
db.getCollection('osmdata').find({"type":"way"}).count()
90327
```

Quantidade de diferentes usuários

- 164

In [ ]:

```
db.getCollection('osmdata').distinct("created.user").length
164
```

Usuários com maior número de contribuições

In [ ]:

```
db.osmdata.aggregate([
{$match: {'created.user': {$exists: 1} }},
 {$group: {_id: '$created.user', count: {$sum: 1}}},
 {$sort: {'count': -1}},
 {$limit: 10}
])
```

In [ ]:

```
{ "_id" : "trilhado", "count" : 560267 }
{ "_id" : "Skippern", "count" : 107209 }
{ "_id" : "BladeTC", "count" : 59262 }
{ "_id" : "LucFreitas", "count" : 40602 }
{ "_id" : "Thundercel", "count" : 5903 }
{ "_id" : "jgiglio", "count" : 2452 }
{ "_id" : "xybot", "count" : 1409 }
{ "_id" : "DenisRizzoli", "count" : 1407 }
{ "_id" : "abel801", "count" : 949 }
{ "_id" : "erickdeoliveiraleal", "count" : 900 }
```

Maiores frequencia de pontos de interesse (amenity):

In [ ]:

```
db.osmdata.aggregate([
{$match: {'amenity': {$exists: 1} }},
 {$group: {_id: '$amenity', count: {$sum: 1}}},
 {$sort: {'count': -1}},
 {$limit: 10}
])
```

In [ ]:

```
{ "_id" : "place_of_worship", "count" : 236 }
{ "_id" : "parking", "count" : 195 }
{ "_id" : "school", "count" : 77 }
{ "_id" : "restaurant", "count" : 57 }
{ "_id" : "bank", "count" : 55 }
{ "_id" : "police", "count" : 46 }
{ "_id" : "fuel", "count" : 39 }
{ "_id" : "fast_food", "count" : 35 }
{ "_id" : "hospital", "count" : 30 }
{ "_id" : "community_centre", "count" : 27 }
```

Conclusões e Melhorias

Após análise dos dados, notou-se a falta de confiabilidade nos dados quanto à quantidade de hospitais encontrados na cidade de vitória (30). Neste número, deve-se considerar postos de saúde, unidades de atendimento ao cidadão, entre outros que se encontram na categoria de hospitais. Decepcionou-se também o número de usuários que contribuem para a manutenção e validação dos dados na cidade Sol.

Solução: O OpenStreet map deveria propor alguns métodos de "gamificação" para incentivar mais usuários a completar o conteúdo os mapas onde residem e, juntamente com os estabelecimentos comerciais e de apoio a população melhorar a base de dados que será de beneficio de todos.

- Beneficios: Adesão de mais usuários para atualizar os dados do OSM - Massificação dos dados
- Problemas esperados: Dependendo do benefício dado pelos estabelecimentos, os usuários cadastrarão dados erroneos em sem padronização, somente para ganhar o tal premio. Com isso teremos mais dados, porém de forma menos filtrada. O desafio será criar um equilibrio entre a massificação de dados com a qualidade deles

Solução: Melhorias no código, seria retirar os "hard codes" de "nodes", "tags" e "way" hoje existentes na função "ShapeElement". Nesse caso, seria interessante, de uma forma mais simples, carregar um vetor com as tags que gostaria que fossem capturadas. Ou, captura-se tudo, e insere no Banco de dados para posterior retirada após análise. Os exercicios encontrados no curso, já "contamina" o aluno a utilizar os valores pré definidos na base de dados. Para uma abordagem em dados "green Field", ou seja, dados que nunca foram explroados, seria interessante ter algumas funções iniciais de exploração para verificar de grosso modo, o que realmente vale a pena explorar.

- Beneficios: As funções poderão ser usadas em outras bases de dados, com necessidade apenas de modificar o dicionario de dados de entrada
- Problemas esperados: Maior complexidade no código, necessitando uma melhor documentação para que o futuro usuário consiga preencher o novo dicionário de dados para capturar as tags

Referencias bibliográficas

A seguir os sites que ajudaram a resolver este projeto:

https://docs.mongodb.com/manual/reference/program/mongoimport/ (https://docs.mongodb.com/manual/reference/program/mongoimport/)

https://www.udemy.com/the-python-mega-course (https://www.udemy.com/the-python-mega-course)

http://jupyter.readthedocs.io/en/latest/index.html (http://jupyter.readthedocs.io/en/latest/index.html)

https://docs.python.org/ (https://docs.python.org/)

http://wiki.openstreetmap.org/wiki/OSM_XML (http://wiki.openstreetmap.org/wiki/OSM_XML)

http://nestacms.com/docs/creating-content/markdown-cheat-sheet (http://nestacms.com/docs/creating-content/markdown-cheat-sheet)