Limpando os dados do OpenStreetMap

Introdução

Esse presente trabalho tem como objetivo, realizar e documentar o processo de limpeza de dados do OpenStreetMap, da cidade de Belo Horizonte. Diante disso, são utilizadas técnicas de tratamento e análise de dados com o intuito de garantir a validade, precisão, consistência e uniformidade dos dados analisados.

A coleta dos dados foi realizada no site do OpenStreetMap, que está disponível no endereço https://www.openstreetmap.org/export, por meio da ferramenta de exportação de dados disponível no próprio site. Essa ferramenta, disponibiliza os dados no formato XML.

Visando facilitar o processo de persistência dos dados coletados no banco de dados MongoDB, optamos por converter o formato dos dados de XML para JSON. Esse processo de conversão é realizado pelo script convertMapXMLToJson.py, que está disponível na pasta src do repositório https://github.com/mrcandrefarias/datawrangling.

A execução do script convertMapXMLToJson.py gera na pasta data um arquivo chamado bh.json, com as informações do mapa de Belo Horizonte no formato JSON. Na sequência, o arquivo mapa.json foi carregado no MongoDB utilizando o comando mongoimport -db bh-osm -c bh --file bh.json, sendo que, bh-osm é nome da base da dados e bh é o nome da coleção.

Problemas encontrados nos dados

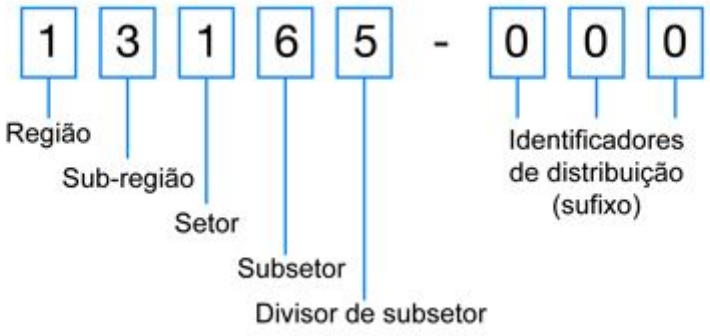
Conforme descrito no próprio site do OpenStreetMap , ele é desenvolvido por mapeadores voluntários que contribuem e ajudam a manter atualizadas as informações sobre estradas, trilhos, cafés, estações ferroviárias entre outras.

De forma geral, a estrutura de coleta de informações é parecida com a da Wikipedia, onde, qualquer pessoa pode editar o mapa, e os próprios colaboradores corrigem os erros uns dos outros, contribuindo para a melhora da qualidade das informações. Com isso, regiões que possuem um número maior de contribuidores, tendem a possuir dados mais precisos.

Durante nossa investigação foi possível observar, erros de digitação, falta de padrões em algumas nomenclaturas e formatações variadas de algumas informações. Após a persistências dos dados no MongoDB, foi possível encontrar os seguintes problemas:

Diferentes formatações de CEP

Por meio de nossas análises, verificamos diferentes formatações de CEP na base de dados da analisada. Carregamos os dados do nosso banco de dados MongoDb e depois disso, enviamos esses dados para a função *is_formato_cep_invalido*. Essa função verifica se o CEP fornecido está formatado conforme especificação dos correios. A imagem abaixo, exibe a estrutura de formatação que está disponível no site dos correios:



Os CEPs fora do padrão de formatação acima, foram trabalhados na função *format_cep_invalido*. A função *format_cep_invalido* realizou a limpeza desses CEPs, eliminando qualquer caractere que não seja número e adicionando o caractere "-" para separar os últimos três números. Feito isso, eles são atualizados no banco de dados MongoDb, com o formato correto.

Logradouros incorretos

Verificamos a ocorrências de muitos nomes de logradouros informados de forma abreviada pelo usuário, como por exemplo exemplo Av e Av. para Avenida e R. para ruas e além disso, alguns desses nomes iniciam com letra minúscula.

A função *verifica_logradouro_invalido* é responsável por verificar e limpar esses erros. Ela valida se o nome do logradouro inicia com letra minúscula e para os casos positivos, ela converte a letra inicial para maiúscula. Além disso, verificamos também se

existe abreviação nos nomes dos logradouros e nos casos afirmativos, optamos por converter os nomes dos logradouro para a forma sem abreviação, conforme listado abaixo:

- Av => Avenida
- Av. => Avenida
- R => Rua
- R. => Rua
- PR => Praça

Após a limpeza dessas informações pela função *verifica_logradouro_invalido*, os logradouros, agora no formato correto, são atualizados no banco de dados MongoDb.

Nome incorreto para a cidade de Belo Horizonte

Realizamos uma consulta em nosso banco de dados para listar os nomes das cidades presentes nos nossos dados, agrupadas pelos seus respectivos nomes. Por meio dessa consulta, foi possível verificar que coletamos dados de outras cidades, além da cidade de Belo Horizonte.

Além disso, observamos que a cidade de Belo Horizonte, foi inserida de diferentes formas pelos colaboradores do OpenStreetMap. Conforme podemos notar abaixo, em alguns casos, existem erros de digitação, como nos casos, *BELO **h**ORIZONTE e belo **H**orizonte* e em outros momentos a cidade é descrita de diferentes formas, como por exemplo *bh, Belo Horizonte/MG* e *Belo Horizonte*.

```
> db.bh.aggregate([ {"$match":{"address.city":{"$exists":1}}}, {"$group": {"_id":"$address.city", "count":{"$sum":1}}}, {"$sort": {"count":-1} }])
{ "_id" : "Belo Horizonte", "count" : 416 }
{ "_id" : "Contagem", "count" : 28 }
{ "_id" : "Betim", "count" : 20 }
{ "_id" : "Esmeraldas", "count" : 11 }
{ "_id" : "Belo Horizonte - MG", "count" : 11 }
{ "_id" : "Sabará", "count" : 8 }
{ "_id" : "Nova Lima", "count" : 7 }
{ "_id" : "Sarzedo", "count" : 4 }
{ "_id" : "Raposos", "count" : 3 }
{ "_id" : "Santa Luzia", "count" : 3 }
{ "_id" : "Ribeirão das Neves", "count" : 2 }
{ "_id" : "Belo horizonte", "count" : 2 }
{ "_id" : "Ibirité", "count" : 1 }
{ "_id" : "BELO hORIZONTE", "count" : 1 }
{ "_id" : "belo Horizonte", "count" : 1 }
{ "_id" : "Belo Horizonte MG Brazil", "count" : 1 }
{ "_id" : "contagem", "count" : 1 }
{ "_id" : "bh", "count" : 1 }
{ "_id" : "belo horizonte", "count" : 1 }
```

{ "_id" : "Santo André", "count" : 1 }
Type "it" for more
> it
{ "_id" : "Belo Horizonte/MG", "count" : 1 }

Devido a esses erros diversos, optamos primeiramente por converter todas as referências à cidade para a forma correta que é Belo Horizonte , conforme pode ser verificado no próprio site da prefeitura do município. A função *verifica_nome_belo_horizonte* é responsável por fazer essa limpeza dos dados e atualizar as informações no banco de dados MongoBD.

Depois disso, realizamos a exclusão de todas as cidades que não fazem referência ao município de Belo Horizonte, uma vez essas outras cidades não fazem parte do escopo de nosso estudo. A função *exclui_cidades_diferentes*, foi responsável pela execução desse trabalho.

Visão Geral dos Dados

Tamanho dos arquivos:

bh.osm: 84 MB

bh.json: 83 MB

Tamanho da base de dados no MongoDB:

bh-osm: 0.029GB

Quantidade de documentos:

> db.bh.find().count() : 421455

Quantidade total de nós:

> db.bh.find({"type":"node"}).count() : 362056

Número de usuários únicos:

> db.bh.distinct('user').length: 663

Top 5 Usuários com mais contribuições:

> db.bh.aggregate([{"$group": {"_id":"$user", "count":{"$sum":1}}}, {"$sort": {"count":-1} }, {"$limit":5}]) :

{ "_id" : "Vítor Dias", "count" : 119869 }

{ "_id" : "Vítor Dias - importação de dados", "count" : 40698 }

{ "_id" : "Gerald Weber", "count" : 35189 }

{ "_id" : "lmpinto", "count" : 23564 }

{ "_id" : "BladeTC", "count" : 22145 }

Número de sinais de trânsito:

> db.bh.find({"highway" : "traffic_signals"}).count(): 898

Número de ciclovias:

> db.bh.find({"highway" : "cycleway"}).count(): 33

Considerações finais

OpenStreetMap é uma poderosa ferramenta colaborativa de mapeamento que tem funcionamento muito parecido com o Wikipedia, onde as informações são criadas e atualizadas por diversas pessoas ao redor do mundo, suas informações são disponibilizadas para uso livre, sendo necessário apenas, creditar a autoria dos dados ao OpenStreetMap e os seus contribuidores.

Apesar de possuir enorme potencial e ter uma base de dados de muito valor, ele ainda não possui muitos contribuidores na cidade de Belo Horizonte e isso tende a prejudicar um pouco a qualidade dos dados, uma vez que, quanto maior o número de contribuidores, melhor será a qualidade das informações. Dessa forma, acreditamos que o aumento do número de contribuidores na região, seria muito importante para garantir maior qualidade e confiabilidade das informações do mapa.

No entanto, acreditamos que apenas o aumento do número de colaboradores não seria suficiente para eliminar todos os erros encontrados no presente estudo. Por isso, a definição de alguns padrões, como por exemplo, padronizar a inserção do CEP conforme orientação dos correios, inserir nomes de avenidas sem o uso de abreviações e inserir o nome da cidade de forma correta (Belo Horizonte), são fundamentais para a qualidade dos dados.

Ainda assim, não seria possível cobrir erros de digitação e outros erros não cobertos no presente estudo. Por isso, o uso de uma ferramenta que permitisse automatizar essas pequenas correções é de extrema importância nesse contexto.

Por fim, para a construção dessa ferramenta, seria necessário um estudo mais aprofundado, capaz de cobrir melhor as informações locais, definir a forma ideal para coleta e envio dos dados automaticamente para o OpenStreetMap e avaliar algoritmos e técnicas de *machine learning* para validação e correção dos dados.

Referências:

OpenStreetMap. Disponível em: http://www.openstreetmap.org/about

Correios. Disponível em: https://www.correios.com.br/para-voce/precisa-de-ajuda/o-que-e-cep-e-por-que-usa-lo/estrutura-do-cep

Prefeitura de Belo Horizonte - Disponível em: https://prefeitura.pbh.gov.br

Expressão regular - Disponível em: https://docs.python.org/2/library/re.html

MongoDB queries - Disponível em: https://docs.mongodb.com/getting-started/shell/query/