P3: Limpando dados do OpenStreetMap

Lieby Cardoso

**Região selecionada para o projeto: Belo Horizonte, MG, Brasil**



Fonte: *http://www.openstreetmap.org/relation/368782*

Conteúdo

[1. Métodos e ferramentas 1](#_Toc470276895)

[2. Problemas encontrados no mapa 1](#_Toc470276896)

[**Acentuação** 1](#_Toc470276897)

[**Vias sem tipo definido** 1](#_Toc470276898)

[2. Visão geral dos dados 1](#_Toc470276899)

[3. Idéias adicionais 2](#_Toc470276900)

[Contributor statistics and gamification suggestion 2](#_Toc470276901)

[Additional data exploration using MongoDB queries 2](#_Toc470276902)

[Conclusion 2](#_Toc470276903)

1. Métodos e ferramentas

A base dos dados explorados neste projeto é disponibilizada pelo OpenStreetMap e foi extraída com a ajuda da ferramenta API Overpass com as coordenadas -44.3092,-20.0979,-43.6116,-19.7376 que representam a área de Belo Horizonte.

O arquivo foi exportado no formato OSM/XML e foi gerada uma amostra desta base com menos informação para facilitar e agilizar a leitura das informações na fase da auditoria. Script executado: Get\_Sample\_OSM.py.

A base (map\_BH.osm) com todas as informações teve seu conteúdo analisado e os apontamentos e decisões tomadas estão documentadas na seção 2 deste documento.

Script executado: Audit\_map\_BH.py.

Auditoria dos dados, o arquivo XML foi convertido para JSON.

Script executado: Dump\_map\_BH\_JSON.py.

Todo o conteúdo do arquivo JSON foi importado para a base BH no banco de dados MongoDB.

Foram executadas consultas aos dados importados e a inconsistências foram corrigidas.

1. Problemas encontrados no mapa

Localização da identificação do tipo de rua

Diferente dos exercícios realizados, onde o mapa era da cidade de Chicago e o tipo da rua podia ser identificado pela última palavra do nome, no mapa de Belo Horizonte, o nome da rua é composto pelo tipo, abreviado ou não, seguido pelo nome. Para atender esta particularidade da nomenclatura brasileira, a variável street\_type\_re foi reformulada para trabalhar com a primeira palavra da frase. A alteração consistiu na inclusão do caracter ^ , que corresponde sempre com o início da string e exclusão do caracter $, que corresponde com o final da string.

street\_type\_re = re.compile(r'^\b(?u)\w\S+\.?', re.UNICODE)

Caracteres especiais

Ao iniciar a auditoria dos dados, imprimi em tela o nome de algumas ruas da amostra e a lógica do programa não foi seguida em alguns casos porque existiam ruas com caracteres especiais como acentos agudos (‘), circunflexos (^), til (~) e cedilha (ç).

No cabeçalho do script foi incluído o comando # -\*- coding: utf-8 -\*- para possibilitar a manipulação destes caracteres com a codificação Unicode utf-8.

Mesmo fazendo essa inclusão, foi necessário acrescentar a letra ‘u’ antes de strings que foram usadas para comparação. Um exemplo é a palavra “Praça” incluída manualmente como string comparativa da variável expected, quando feita a comparação de “Praça” digitado com Praça retornando de uma variável, a comparação não era bem sucedida, neste caso a palavra “Praça” digitada foi substituída por u”Praça” para ser reconhecida como contendo caractere especial.

expected = ["Rua", "Avenida", "Beco", "Rodovia", "Expressa", u"Praça", "Anel", "Alameda"]

mapping\_street = { "Av.": "Avenida",

"Av": "Avenida",

"Rod.": "Rodovia",

"Al.": "Alameda",

"R.": "Rua",

u"Anél": "Anel"}

Outro tratamento foi incluir a codificação (?u)\w para todos os caracteres Unicode no re.compile.

street\_type\_re = re.compile(r'^\b(?u)\w\S+\.?', re.UNICODE)

Vias sem tipo definido

Algumas vias não tinham o seu tipo definido, como por exemplo, avenida, alameda, beco e etc.. Quando a via não tinha o tipo compatível com os cadastrados no array expected, nem no dicionário mapping\_street, atribuí o prefixo “Rua” para todas elas.

Montes Claros => Rua Montes Claros

Av São Lucas => Avenida São Lucas

Av Afonso Pena => Avenida Afonso Pena

Av Dom Pedro II => Avenida Dom Pedro Ii

Anél Rodoviário => Anel Rodoviário

rua das orquídeas => Rua Das Orquídeas

Paracatu => Rua Paracatu

Avendia Dom Pedro II => Avenida Dom Pedro Ii

Alamedas das Princesas => Alameda Das Princesas

RUA LUZIA SALOMÃO => Rua Luzia Salomão

Eua Conceição Aparecida => Rua Conceição Aparecida

Paraíba => Rua Paraíba

Pc Nossa Senhora do Rosário => Praça Nossa Senhora Do Rosário

Av. Dom Pedro II, => Avenida Dom Pedro Ii,

Av. Alcides Fernandes de Souza => Avenida Alcides Fernandes De Souza

Av. Antônio Carlos => Avenida Antônio Carlos

Av. Abraao Caran => Avenida Abraao Caran

Av. Bias Fortes => Avenida Bias Fortes

Av. Dom José Gaspar => Avenida Dom José Gaspar

Francisco Deslandes => Rua Francisco Deslandes

Riachuelo => Rua Riachuelo

Av.Afonso Pena => Avenida Afonso Pena

Pirité => Rua Pirité

R. de Áries, => Rua De Áries,

R. Prof. Melchíades da Costa Lage => Rua Prof. Melchíades Da Costa Lage

R. A => Rua A

R. das Contendas => Rua Das Contendas

avenida dom José Gaspar => Avenida Dom José Gaspar

Pium-I => Rua Pium-I

R.B => Rua B

R. Prof. Melchíades da Costa Lage => Rua Professor Melchíades Da Costa Lage

1. Visão geral dos dados
   1. Representação dos dados OSM

Os dados estão organizados em uma estrutura de elementos do mapa que representam o objeto e seu valor associado. Tipos encontrados:

node: um nó é informações de um par de coordenadas, sendo latitude e longitude;

way : uma lista de nós (nodes);

area: representa uma área limitada, por exemplo, uma praça;

relation: pode ser um conjunto de nodes, ways e relation relacionados.

File sizes

charlotte.osm ......... 294 MB

charlotte.osm.json .... 322 MB

# Number of documents

# Number of nodes

# Number of ways

# Number of unique users

# Top 1 contributing user

# Number of users appearing only once (having 1 post)

# “\_id” represents postcount

1. Idéias adicionais

Contributor statistics and gamification suggestion

Additional data exploration using MongoDB queries

# Top 10 appearing amenities

# Biggest religion (no surprise here)

# Most popular cuisines

1. Conclusion

After this review of the data it’s obvious that the Charlotte area is incomplete, though I believe it has been well cleaned for the purposes of this exercise. It interests me to notice a fair amount of GPS data makes it into OpenStreetMap.org on account of users’ efforts, whether by scripting a map editing bot or otherwise. With a rough GPS data processor in place and working together with a more robust data processor similar to data.pyI think it would be possible to input a great amount of cleaned data to OpenStreetMap.org.

1. Referências

http://wiki.openstreetmap.org/wiki/Belo\_Horizonte

<http://wiki.openstreetmap.org/wiki/WikiProject_Brazil>

http://overpass-api.de/api/map?bbox=-44.3092,-20.0979,-43.6116,-19.7376