Tutorial Introdutório ao Google Refine

Flávio Codeço Coelho Renato Rocha Souza

2 de abril de 2012

Resumo

Neste tutorial vamos conhecer o Google Refine[1] e aprender a como utilizá-lo na limpeza de conjuntos de dados mal-estruturados e com vários tipos de problemas.

[1] `http://code.google.com/p/google-refine`

Sumário

Capítulo 1

Introdução

1.1 Visão Geral

O Google Refine é uma ferramenta para lidar com dados de má qualidade, voltada principalmente para ajudar a localiza e consertar inconsistências.

Capítulo 2

Importando Dados

Para poder analisar dados no Refine é preciso primeiro importá-los. O Refine sabe como ler arquivos nos seguintes formatos:

- TSV, CSV, ou valores separados por um separador especificado por você.
- Excel (.xls, xlsx)
- XML, RDF com XML
- JSON
- Google Spreadsheets
- RDF triplas N3

O Refine também importa multiplos arquivos compactados com extensões .zip, .tar.gz, .tgz, .tar.bz2, .gz, or .bz2. Neste caso o Refine detecta o tipo de arquivo mais comum e importa todos os dados segundo este formato.

Capítulo 3

Explorando os Dados

A primeira etapa em qualquer tarefa de análise de dados consiste e nos familiarizar com o conjunto de dados. Para este fim, o Refine oferece algumas ferramentas de exploração de dados que vamos apresentar a seguir.

3.1 Facetação

Facetação refere-se a um conjunto de técnicas que visam resumir um conjuntos por meio de agregação baseada no valor das células de uma coluna.

O tipo mais comum de facetação é por texto. Por exemplo, na tabela de andamentos do Supremo em Números, podemos facetar pela coluna andamento (figura 3.1). Ao fazer isso, temos como resposta um agrupamento pelos valores destas colunas. A esquerda da tela vemos todas as strings distintas que esta coluna apresenta, e podemos então clicar em uma restringir avisualização dos dados a apenas as linhas que possuem aquela string na coluna "andamento" (veja figura 3.1).

O refine possui vários tipos de rotinas de facetação pré-programadas (por texto, número, datas, numérica, etc., ver figura 3.2) e ainda permite que programemos uma facetação arbitrária.

Ainda podemos facetar por múltiplas colunas por meio de programação:

```
cells["andamento"].value[0] == cells["observacao"].value
    [0]
```

O exemplo acima demonstra como podemos facetar agrupando linhas em que a primeira letra é igual de duas colunas de texto é igual.

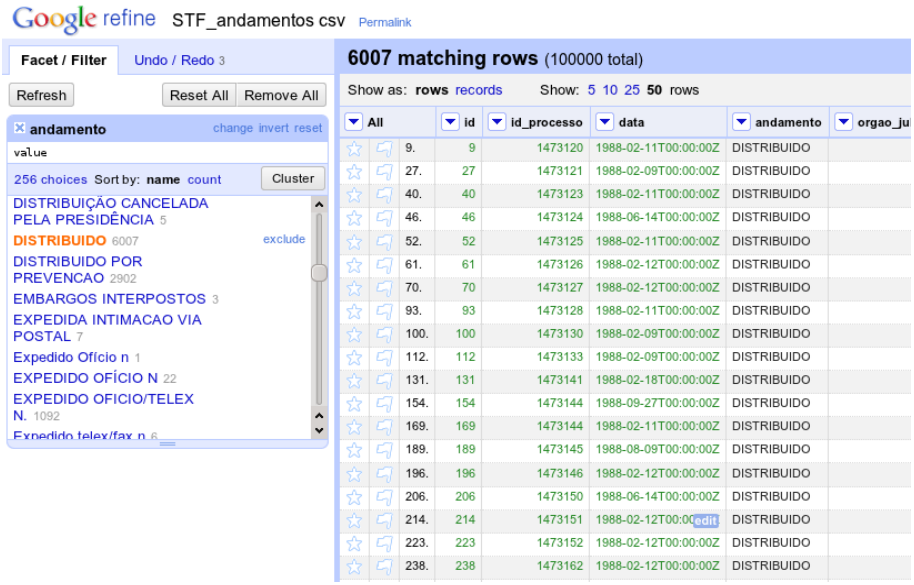

Figura 3.1: Facetação simples por texto.

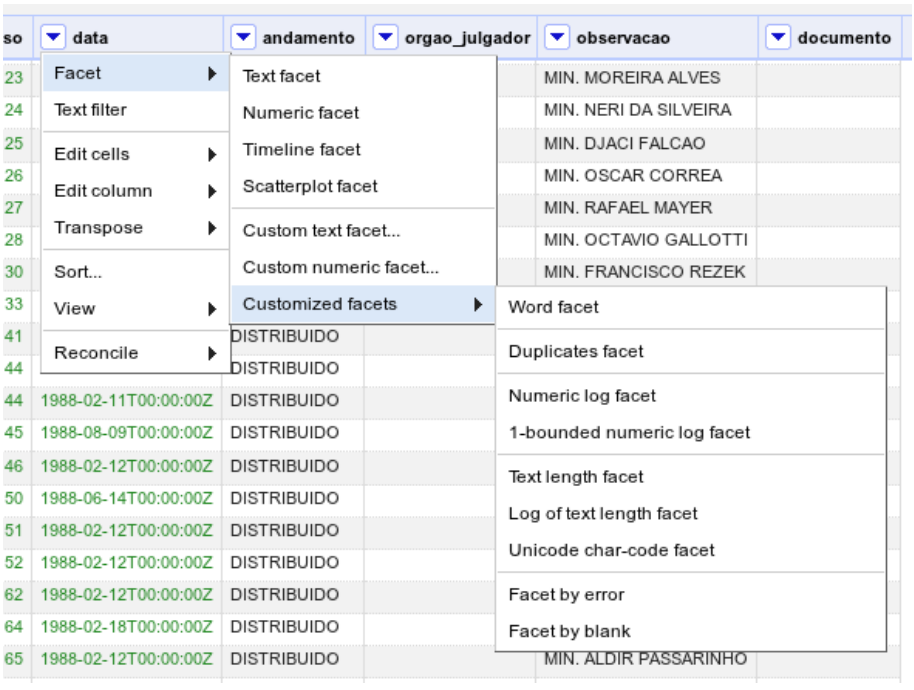

Figura 3.2: Tipos de facetação oferecidos pelo Refine

Capítulo 4

Transformando dados

4.1 Transformando Texto

4.1.1 Mascarando dados sensíveis

Muitas vezes precisamos trabalhar com dados que são confidenciais ou não podem ser revelados a quem precisa analisá-los. Uma técnica matemática muito eficaz para mascarar textos de forma concisa é a transformação do dado por meio de uma função de embaralhamento (hash) criptográfico.

Segundo a definição da Wikipedia[1]:

> A Função de embaralhamento criptográfico é uma procedimento determinístico que leva a um bloco arbitrário de dados e devolve uma cadeia de caracteres de bits com tamanho fixo, o valor (criptográfico) de embaralhamento, de tal forma que uma mudança acidental ou intencional de dados irá alterar o valor do embaralhamento (ver figura 4.1). Os dados a serem codificados muitas vezes são chamados de "mensagem", e o valor de embaralhamento é chamado às vezes de a resumo da mensagem ou simplesmente resumo.

Usando a linguagem Python, podemos pedir ao Google refine para substituir uma coluna inteira por sua versão embaralhada pela função MD5.

```
import md5;
hash=md5.md5(value)
return hash.hexdigest()
```

Veja na figura 4.2 como implementar esta transformação no Refine. Após selecionar no menu da coluna que se deseja transformar, a opção "edit cell" | "transform...", copie o código Python acima conforme ilustrado na figura 4.2, não esquencendo de selecionar Jython no menu "language". Finalmente pressione "OK" para aplicar o resultado aos seus dados.

4.1.2 Limpando Dados

Imagine que tenhamos uma coleção de endereços como a mostrada na figura 4.3. Note que na coluna endereço, temos uma mistura de logradouro, com número e

[1] http://pt.wikipedia.org/wiki/Função_de_embaralhamento_criptográfico

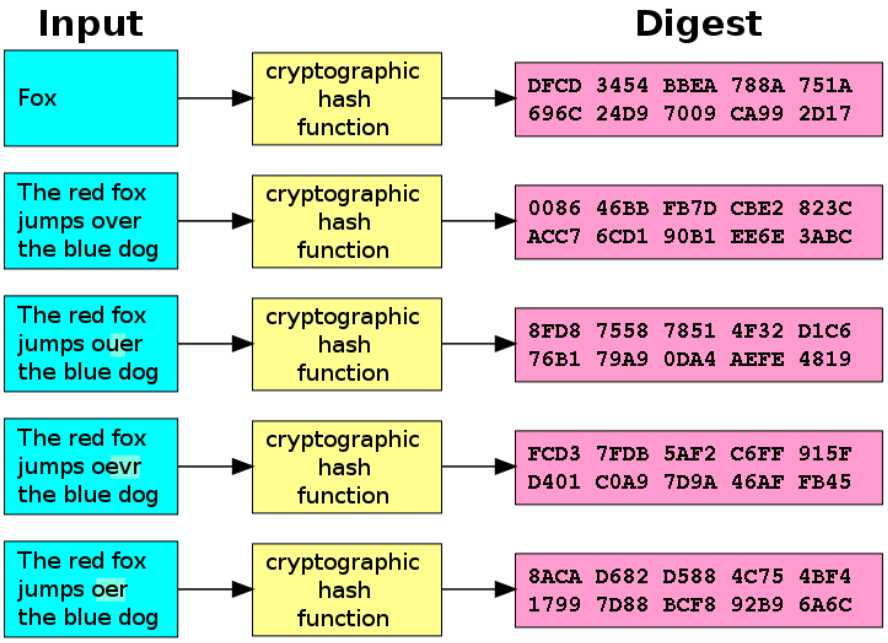

Figura 4.1: Uma função criptográfica de embaralhamento (especificamente, SHA-1) funcionando. Note que mesmo pequenas mudanças no código-fonte alteram drasticamente a saída do resultado, pelo chamado efeito avalanche.fonte: Wikipedia

salas. Este formato é pouco interessante se desejarmos comparar estes endereços com uma base canônica de endereços, como a dos correios, por exemplo. Vamos ver então como podemos extrair apenas o logradouro e número destes endereços formatados livremente. Como vamos realizar uma edição complexa de texto, convém antes transformar todas as células para caixa alta. Podemos fazer isso com a função "edit cells" → "common transforms" → "to uppercase".

No menu da coluna endereço vamos selecionar a opção "edit column" → "add column based on this column"

Listing 4.1: codigo Python para limpeza dos endereços

```
nv = value.replace(', N.   ',',').split('/')[0]
nv = nv.replace('N  ',',')
nv = nv.replace('N.   ',',')
nv = nv.replace(', N.',',')
nv = nv.replace('N.',',')
nv = nv.replace(', ,',',')
nv = nv.split('-')[0]
nv = nv.split('SALAS')[0]
nv = nv.split('SALA')[0]
nv = nv.split('SLS')[0]
return nv
```

A operação de limpeza exemplificada na listagem 4.1, não é perfeita, mas já

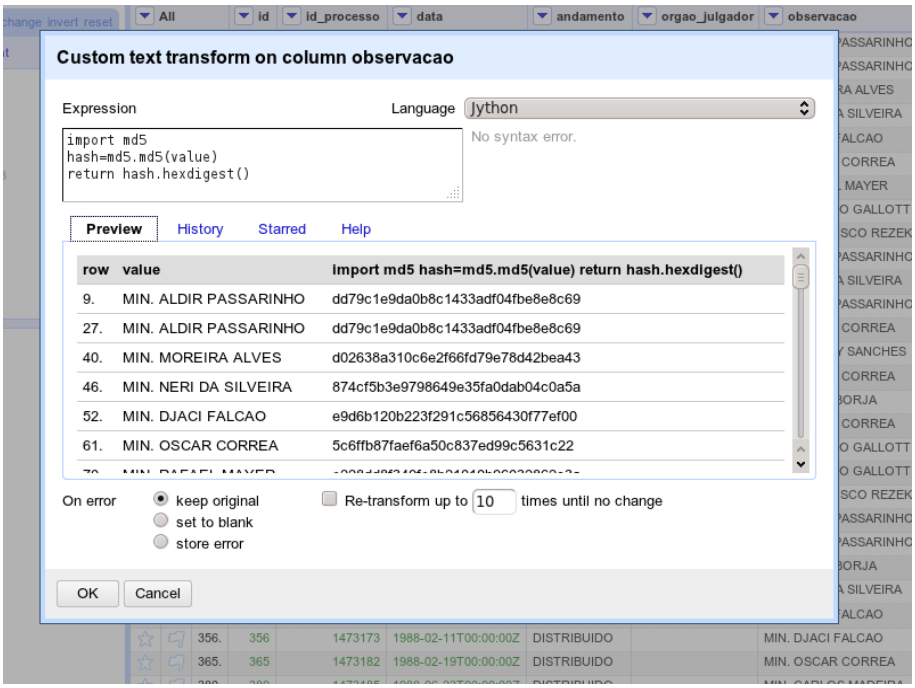

Figura 4.2: substituindo o valor de uma coluna pelo seu MD5 hash

melhora bastante a qualidade do nosso campo endereço. A figura 4.4, mostra-nos como implementar a limpeza.

60 matching rows (113 total)

Show as: **rows** records Show: 5 10 25 **50** rows

	orgao	numero	endereco	cidade	bairro	cep
			AV NILO PEÇANHA, 11 SALAS 603/604	RIO DE JANEIRO	CENTRO	
			AV. PAULO DE FRONTIN 590, SLS. 804 A 809	VOLTA REDONDA	ATERRADO	
			Av. Nilo Peçanha nº 151/ gr 211	RIO DE JANEIRO	CENTRO	
			Rua da Assembléia n.º 58/11º andar	RIO DE JANEIRO	CENTRO	
			AV. BEIRA MAR, 216 - 3º ANDAR	RIO DE JANEIRO	CENTRO	
			AV. PRESIDENTE WILSOM, 165 - SALA 913	RIO DE JANEIRO	CENTRO	
			Av. Almirante Barros nº 63/1403 e 1404	RIO DE JANEIRO	CENTRO	
			AV. ALMIRANTE BARROSO n. 139 / 5º ANDAR	RIO DE JANEIRO	CENTRO	

Figura 4.3: Endereços sem normalização

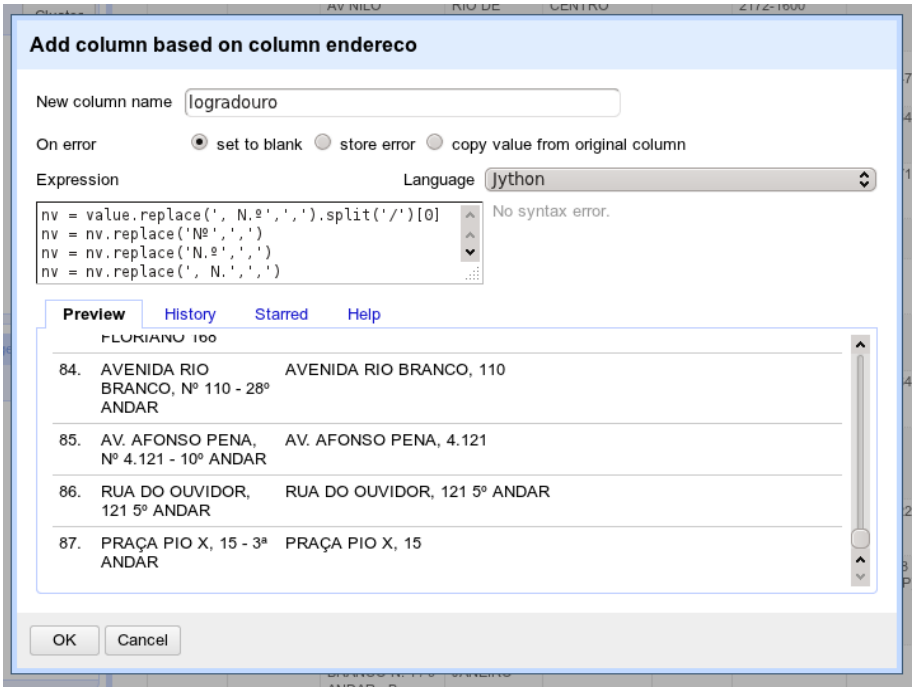

Figura 4.4: Criando uma nova coluna "logradouro"baseada na limpeza da coluna endereço.

Capítulo 5

Enriquecendo Dados

Neste capítulo vamos lidar com técnicas conhecidas no jargão técnico com "data augmentation". Estas técnicas consistem em capturar dados de outras fontes de forma a enriquecer nosso conjunto original de dados. Esta não é uma tarefa simples porém pois os novos dados devem ser pareados com os dados pré-existentes.

5.1 Capturando Dados a partir da Web

Suponhamos que nós temos em uma coluna do nosso banco de dados, endereços. Endereços são um tipo de dado que nos fornece informação com relação à localização espacial de nosso dados mas, se quisermos construir uma visualização espacial destes dados precisaremos de informações mais específicas, como a latitude e longitude destes endereços.

Como podemos tentar resolver este problema no Google Refine?

Na seção 4.1.2, realizamos a limpeza de um conjunto de endereços. Para obter a geolocalização destes endereços a partir de uma fonte aberta de dados, podemos usar a api do Openstreetmap[1] (para maiores informações leia documentação[2])

Para criar uma coluna com as coordenadas dos endereços, precisamos selecionar *"add column by fetching URLs based on column endereço"*.

então digitamos a url que desejamos consultar. No nosso casos é esta.

```
'http://nominatim.openstreetmap.org/search?format=json&
    email=seuemail.gmail.com&q=' + escape(value,'url') + '
    ,' + escape(cells['cidade'].value,'url')
```

temos que configurar o campo throttle para no mínimo 1500 milisegundos para evitar recusas no servidor do Openstreetmap. Neste caso utilizaremos a linguagem GREL (Google Refine Expression Language). Vamos nomear a nossa nova coluna de json, pois os dados que vamos receber estão em formato JSON[3] Agora clicamos *Ok* e aguardamos o resultado(figura 5.1). Depois de recebermos os endereços em formato JSON, Necessitamos extrair apenas os dados que queremos: Latitude e Longitude. Para isso vamos usar a opção "add

[1] `http://nominatim.openstreetmap.org`
[2] `http://wiki.openstreetmap.org/wiki/Nominatim`
[3] JSON: JavaScript Object Notation

column based on this column" da coluna json. Na janela de criação da coluna adicionaremos o seguinte código em GREL:

Listing 5.1: Extraindo latitudes e longitudes dos endereços em JSON

```
with(value.parseJson()[0],par,par.lat+','+par.lon)
```

Figura 5.1: Resultado da busca no OpenStreetMap

A figura 5.2, mostra a extração dos valores por meio do código da listagem 5.1.

Agora que dispomos das latitudes e logitudes, podemos exportar nossa tabela e preparar uma visualização dos nossos dados usando, por exemplo, outra ferramenta gratuita, o Google Fusion Tables(figura 5.3).

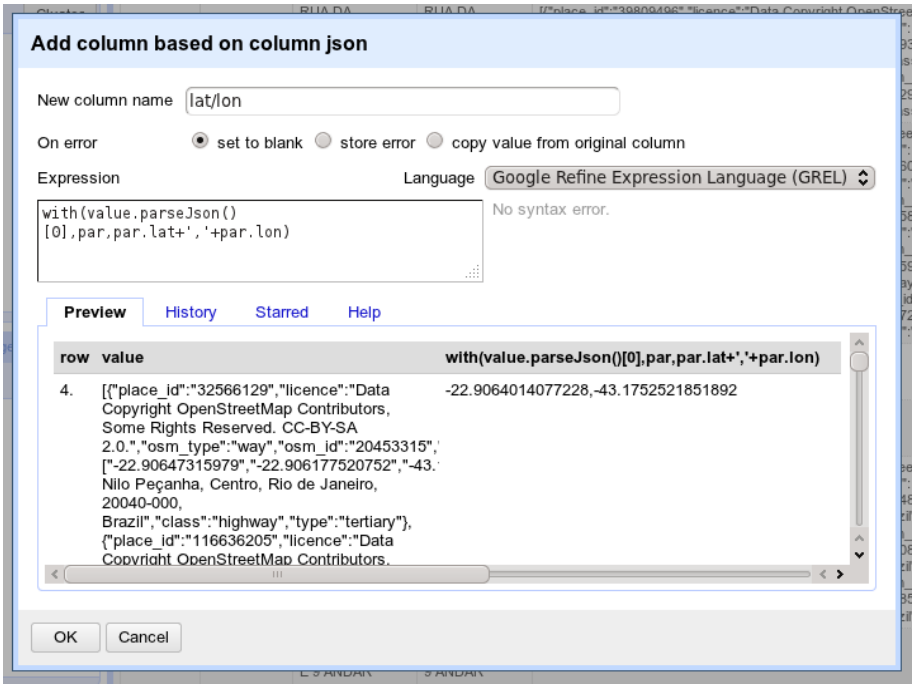

Figura 5.2: Extraindo latitudes e longitudes dos endereços em JSON

Figura 5.3: Visualização dos endereços geo-localizados no Fusion Tables