Projeto Final MegaDados

Insper

Prof. Fábio Ayres

Membros: Martim F., Sabrina S., Leonardo M

Proposta

A principal objetivo deste projeto era manipular e analisar um grande conjunto de dados, considerados Big Data. Para isso, foi implementado uma Pipeline em um EMR (Elastic Map Reduce) da AWS que utilizou por meio do Zeppelin o Pypark, para manipular os dados do Common Crawl (http://commoncrawl.org/) do mês de Setembro, que compreende terabytes de dados, da Web mundial. Porém como o professor já havia feito um filtro dos sites brasileiros, esses terabytes de dados se transformaram em 64GB salvos em um bucket S3.

A partir dos dados, analisamos estatisticamente a frequência das palavras que são ditas em conjunto com os nomes das capitais dos estados do Brasil. Para assim mapear o vocabulário associado a cada capital dos estados.

Obtenção dos Dados

Para obter os dados da web brasileira, acessamos os dados do bucket (s3://megadados-alunos/data/web_brasil/data/web-brasil) criado pelo professor, que continha 297659 sites, totalizando 64GB armazenados em um Python RDD (Resilient Distributed Dataset), facilitando a manipulação em cluster.

Com isso, iteramos por todos os documentos no RDD (compostos por uma tupla, que armazena a URL e o conteúdo do site) varrendo o conteúdo e contando a frequência com que cada palavra aparece, armazenando essa contagem em um dicionário. Ao fim, as palavras mais frequentes na web brasileira foram:

```
[('de', 11752063), ('e', 6631954), ('a', 4637523), ('do', 3707632), ('o',
 3443619), ('em', 2532392), ('para', 2430488), ('da', 2393455), ('1', 227
7141), ('com', 2113162)]
```

Esses dados foram salvos em um pickle '/tmp/frequencia_palavras_geral.pickle' para análise póstuma.

Para fazer a mesma contagem para as palavras que são ditas em conjunto com as capitais de cada estado, foi construído um dicionário a partir das capitais e feita a contagem das palavras para todos os documentos que a citam. Para a chave São Paulo por exemplo, obteve-se essa lista de tuplas:

```
[('de', 4683738), ('e', 2730197), ('a', 2173891), ('do', 1852412), ('o',
 1185111), ('da', 1061440), ('em', 1055275), ('para', 980817), ('1', 9437
55), ('com', 826408)]
```

Esses dados foram salvos em um pickle '/tmp/frequencia_palavras.pickle' para análise póstuma.

Como estávamos executando essa pipeline em um servidor AWS, para acessar os pickles precisamos acessar via SSH a máquina e fazer o upload dos arquivos para um bucket do S3 via: `aws s3 cp frequencia_palavras.pickle s3://br-web-crawler/frequencia_palavras`

Tradução do Contexto em Hipóteses

Com isso, pensamos em analisar a frequência de palavras em documentos e montar um bag of words (BOW) para a totalidade dos documentos. Sequencialmente, fizemos um filtro que pegava apenas os documentos em que o nome de cada um dos 26 estado do Brasil fosse mencionado, e montamos um BOW individual para cada.

Será que uma palavra (ω) é mais frequênte em textos em que "São Paulo" é mencionado? Por exemplo... "Bras" Será que as preposições são mencionadas igualmente no BOW total e nos BOWs filtrados? Provavelmente "Tu" para "São Paulo" terá uma frequência bem menor que do total.

Assim, temos:

> Hipótese 0: freq ω (estado) = freq ω (total)

Em contrapartida, é possível que palavras tenham frequências totalmente divergêntes entre um estado e a visão geral dos documentos coletados:

> Hipótese 1: freq ω (estado) ≠ freq ω (total)

Mecanismos para Cálculo do p-value

Foi calculado um pvalue experimental de cada palavra em seu respectivo BOW, usando como parâmetro o teste de hipotese para distribuição binomial.

$$\hat{p} = \frac{n_t p_t + n_e p_e}{n_t + p_e}$$

$$Z = \frac{p_e - p_t}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{n_e} + \frac{1}{n_t})}}$$

Onde:

n_t é a quantidade da palavras no dicionario total

n_e é a quantidade da palavras no dicionario do estado

p_t é a fracao da palavra por todas as palavras no dic total

p_e é a fracao da palavra por todas as palavras no dic do estado

Usando a biblioteca do scipy, calcula-se a distribuição cumulativa de uma normal passando como parâmetro o valor Z

Assim, temos:

$$Pvalue = (1 - norm.cdf(Z)) * 2$$

Estruturação dos Dados Coletados Para Analise

Em posse dos arquivos de contagem das palavras, foi realizado duas estruturações.

A primeira estruturação consiste em tratar os dados coletados e calcular o pvalue de cada palavra e armanena-los em um dicionario, no qual cada chave, que representa a capital, possui uma lista de tuplas, cada tupla comtem as informações de uma palavra, ordenada da seguinte forma: "palavra", "porcentagem na capital", "porcentagem no estado", "pvalue", toda esta estapa é realizada pelo script br-web-crawler.py

A segunda etapa de estruturação consiste em carregar o arquivo gerado na primeira etapa, que contem um dicionario, e transforma-lo em um dataframe, formato necessario para realizar a analise dos dados, o dataframe é filtrado, mantendo apenas as palavras que comtem valor de pvalue menor que 0.05, com a finalidade de manter apenas os valores estasticamente significativos, posteriormente o dataframe é apresentado em uma tabela com seus elementos ordenados de forma descendente pela razão da fração da palavra no estado e a fração da palavra na web geral, toda esta etapa e realizada pelo script MegaDadosFinal.ipynb

Resultados