

# Projeto Final MegaDados

---

Inspere

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 Pyspark, 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](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', 2277141), ('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', 943755), ('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 estados do Brasil fosse mencionado, e montamos um BOW individual para cada.

Será que uma palavra ( $\omega$ ) é mais frequente em textos em que "São Paulo" é mencionado? Por exemplo... "Bicicleta". 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:  $\text{freq } \omega \text{ (estado)} \neq \text{freq } \omega \text{ (total)}$

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

Hipótese 1:  $\text{freq } \omega \text{ (estado)} = \text{freq } \omega \text{ (total)}$

## Mecanismos para Cálculo do p-value

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

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

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

Onde:

$n_t$  é a quantidade das palavras no dicionário total

$n_e$  é a quantidade das palavras no dicionário da capital

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

$p_e$  é a fração da palavra por todas as palavras no dic da capital

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

Assim, temos:

$$Pvalue = (1 - \text{norm.cdf}(Z)) * 2$$

## 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 etapa 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 contem 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

Os resultados a seguir são dataframes onde os 10 primeiros itens são teoricamente os mais falados em documentos que se menciona uma capital (algumas colunas de certas dataframes foram cortadas por apresentar dados irrelevantes e muito poluídos).

São Paulo						Palmas					
	palavra	P_total	P_estado	P-value	Fração		palavra	P_total	P_estado	P-value	Fração
1353	Brás	0.000438	0.000925	0.000000e+00	2.111585	597	Hino	0.000060	0.000729	6.436526e-09	12.054512
1713	Plástica	0.000183	0.000380	4.067096e-04	2.081623	405	celular	0.000049	0.000323	3.229839e-03	6.635018
3311	Cirurgia	0.000190	0.000395	2.427129e-04	2.077742	80	cursos	0.000053	0.000288	8.316691e-03	5.383889
1243	Atacado	0.000244	0.000506	2.695500e-06	2.075090	494	Cidades	0.000041	0.000220	4.389830e-02	5.313249
7475	Moda	0.000603	0.001190	0.000000e+00	1.973552	665	PR	0.000066	0.000234	4.952367e-02	3.551968
5876	Paulo	0.000352	0.000683	2.258746e-09	1.937203	818	Jardim	0.000119	0.000407	7.727734e-04	3.420951
1908	Melhor	0.000198	0.000383	8.175010e-04	1.935606	925	MG	0.000086	0.000293	1.541368e-02	3.418939
8742	Feminina	0.000125	0.000239	3.846649e-02	1.912266	763	Vila	0.000098	0.000328	7.302606e-03	3.348916
244	Lojas	0.000146	0.000264	3.093090e-02	1.810980	870	do	0.003739	0.002318	0.000000e+00	0.619912
3849	Festa	0.000160	0.000287	1.972048e-02	1.800170	151	de	0.011852	0.006342	0.000000e+00	0.535067

Boa Vista				Brasília				Campo Grande				Florianópolis			
	palavra	P_total	P_estado		palavra	P_total	P_estado		palavra	P_total	P_estado		palavra	P_total	P_estado
1098	Hino	0.000060	0.000516	4292	Brasilia	0.000042	0.000168	1127	Hino	0.000060	0.000497	1554	Florianópolis	0.000024	0.000176
734	celular	0.000049	0.000237	3777	RO	0.000027	0.000083	172	rastrear	0.000011	0.000089	265	rastrear	0.000011	0.000075
912	Buffet	0.000082	0.000398	4625	LIVRARIA	0.000045	0.000133	214	Montador	0.000009	0.000074	1404	Hino	0.000060	0.000412
130	cursos	0.000053	0.000221	1419	SHOPPING	0.000049	0.000139	564	Desentupidora	0.000016	0.000123	63	Hinos	0.000019	0.000117
1497	Jardim	0.000119	0.000470	2150	DF	0.000029	0.000082	70	Hinos	0.000019	0.000140	79	BRL	0.000020	0.000104
1420	Vila	0.000098	0.000381	786	CULTURA	0.000051	0.000136	1385	st	0.000032	0.000201	2025	Salvar	0.000018	0.000089
239	Infantil	0.000177	0.000463	3498	Hino	0.000060	0.000145	13	BRL	0.000020	0.000122	1562	SC	0.000056	0.000229
1069	07	0.000294	0.000641	2906	mapa	0.000035	0.000079	445	gratis	0.000013	0.000080	934	celular	0.000049	0.000189
871	08	0.000329	0.000653	2284	celular	0.000049	0.000110	602	prende	0.000018	0.000097	1112	Catarina	0.000027	0.000089
1090	05	0.000296	0.000569	460	cursos	0.000053	0.000112	727	celular	0.000049	0.000232	1666	↳	0.000035	0.000114

Contudo, algumas capitais tiveram o infortúnio de preencherem o BOW com código HTML mal formatado, como aconteceu claramente com o Rio de Janeiro. O BOW ficou extremamente poluído e não trouxe grandes adições ao projeto.

Dessa forma, algumas capitais não tiveram suas dataframes exibidas.

Rio de Janeiro					
	palavra	P_total	P_estado	P-value	Fração
2945	subHome	0.000106	0.000337	3.757705e-04	3.183242
3651	component	0.000323	0.001028	0.000000e+00	3.179757
3086	ig	0.000153	0.000481	4.014907e-07	3.139452
7878	nav	0.000140	0.000426	8.344309e-06	3.051376
2094	10px	0.000062	0.000189	4.789582e-02	3.038676
6837	block	0.000079	0.000239	1.281983e-02	3.009703
1093	barra	0.000081	0.000243	1.127689e-02	3.005114
1143	lateral	0.000072	0.000215	2.542946e-02	2.980549
2600	bottom	0.000107	0.000318	9.416578e-04	2.976774
7807	font	0.000160	0.000474	8.648709e-07	2.959840

## Conclusão

Dados os resultados acima, não podemos rejeitar a hipótese nula de que as palavras tem frequências diferentes entre total e estado, pois os gráficos indicam valores de palavras bem mais populares para certas capitais. Em São Paulo, "Brás" é um campeão. Já em Boa Vista, temos "Buffet".

Contudo, há uma situação de fronteira que deve ser explicitada que são as bag of words que contém muito texto quebrado HTML ou muitas preposições. Neste caso, não é possível nem rejeitar nem aceitar H0, visto que os dados são irrelevantes e inoperáveis.

Basicamente, cada capital resultou em uma tabela muito interessante que expõe as palavras mais relacionadas a si, e apesar de haver certas falhas operacionais, o resultado é algo intrigante que pode ser expandido e explorado.

## Melhorias Necessárias

Um tópico que não pôde ser coberto direito foram as palavras menos relacionadas com cada capital, ou seja, a frequência naquele lugar é menor que do BOW total. Alguns exemplos foram gerados, mas todos poluídos por símbolos, HTML e preposições. Um breve exemplo das 10 palavras menos relacionadas com São Paulo.

<b>Menos Relacionadas com SP</b>
----------------------------------

Menos Relacionadas com SP					
	palavra	P_total	P_estado	P-value	Fração
7634	'	0.000308	0.000117	1.374083e-02	0.379525
2883	►	0.001158	0.000592	2.220446e-16	0.511442
1165	Ver	0.000339	0.000203	3.428226e-02	0.597132
6086	component	0.000323	0.000194	4.490833e-02	0.599998
1858	t	0.000584	0.000353	3.272612e-04	0.604505
4450	2017	0.000414	0.000260	1.579257e-02	0.629874
3538	2016	0.000350	0.000226	4.880847e-02	0.646051
9269	R	0.001359	0.000888	5.040413e-14	0.653092
672	00	0.000622	0.000407	5.911799e-04	0.654114
6061	5	0.000619	0.000410	8.207331e-04	0.663142

Além disso, poderíamos classificar as páginas da web brasileira por meio da linguagem, porque a atual classificação foi feita por meio do domínio .br, portanto deixou de lado muitas outras páginas brasileiras que não utilizam esse domínio, como o site de notícias [www.globo.com](http://www.globo.com).

Um último aspecto que poderia ser melhorado é a pipeline no momento em que se cria o BOW. Ao invés de contar todas as vezes que uma palavra aparece em um documento (para todos os documentos), poderia-se contar apenas uma vez por documento. Deste modo, acredita-se que os resultados saíam melhores, tendo