Análise descritiva
Transparência em Tempos de Pandêmia

Gabriel Martins Trettel - N. USP 11389471
George Othon Silva Santos - N. USP 10349978
Tiago Lubiana Alves - N. USP 8945857
Wesley Seidel Carvalho - N. USP 6544342

São Paulo, 08 de Setembro de 2020

1 Descrição inicial da proposta

Neste trabalho descrevemos o entendimento inicial do problema apresentado pelo cliente Open Knowledge Brasil (OKB) tendo como contato o sr. Mário. Também descrevemos os dados que nos foram apresentados e a forma com a qual foram obtidas pelo cliente. O trabalho diz respeito à análise de dados textuais extraídos de diversos diários oficiais.

Uma primeira reunião com o cliente foi realizada no dia 04/09/2020, na qual estiveram presentes o sr. Mário Queiroz e Ariane representando a OKB e a equipe de desenvolvimento deste trabalho. Após conversa com os representantes da OKB, começamos a definir os objetivos do projeto.

Devido a riqueza dos dados e a natureza múltipla dos interesses do cliente, pautamos um objetivo inicial para orientar a exploração dos dados: Um dos desejos citados pela equipe da OKB é que em algum momento, os interessados em informações dos Diários Oficiais consigam identificar compras suspeitas durante a pandemia. Neste caso, a tarefa consistirá tanto em identificar compras e contratações feitas em determinado período quanto em organizar tais informações e possibilidades de integração com bases de dados sugeridas pelo cliente, tais como a base de sócios de empresas brasileiras e a base de dados de doações de campanha do Tribunal Superior Eleitoral.

Tal tarefa implica, à priori, extração de informações dos diários tais como: valores envolvidos em transações; partes envolvidas nas transações; data das transações. A partir disso, então, será possível pensar em estratégias para identificar situações que possam ser consideradas suspeitas, a exemplo de donos de empresas envolvidas tanto com financiamento de campanhas nas eleições de 2020 como em transações comerciais relatadas em diário oficial.

Algumas palavras sobre a exploração dos dados piloto dos Diários Oficiais: Os dados foram fornecidos em diversas pastas, com categorias distintas. Os arquivos estão presentes em formato txt, com diversos arquivos por pasta. Neste documento realizamos algumas análises iniciais sobre esta massa de dados além da discussão de algumas possibilidades de implementação que nos auxilie em definir objetivos mais claros e auxilie o cliente no vislumbre de um produto viável.

1.1 Descrição da organização dos dados

A descrição dos dados, nos fornecida pelo Mario Sérgio, nosso ponto de contato da *Open Knowledge Foundation*, é a seguinte:

Foram coletados arquivos em PDF de 306 municípios brasileiros, listados em um repositório no *GitHub* da organização. O intervalo de publicação dos diários é do dia 01 de fevereiro de 2020 até o dia 15 de junho de 2020. Os arquivos foram convertidos de PDF para txt e, então, uma busca por palavras-chave foi exexutada para tentar separa-los por assunto. As palavras-chaves utilizadas foram:

- Emergencial

- Estado de Emergência de Saúde Pública
- Dispensa de licitação
- Equipamentos de Proteção Individual
- EPI
- Ventiladores pulmonares
- Ventilador pulmonar
- Demanda Emergencial Covid-19
- Teste rápido
- RT-PCR
- Hospital de Campanha

Pela natureza do que foi nos dado, não temos de maneira direta a informação dos municípios do qual cada diário foi obtido, nem suas respectivas datas de publicação. Iremos tentar obter essa informação para as etapas seguintes do processo, após interagir com a equipe responsável pelo *scraping* das páginas dos municípios.

2 Descrevendo os arquivos apresentados

Tendo em vista que os dados foram provenientes de buscas por palavras-chave, decidimos ver o número de arquivos para cada busca, visto na tabela 1:

Termo de busca	Qnt	(MB)
Demanda Emergencial Covid-19	228	109M
Dispensa de licitação	800	290M
emergencial	440	198M
EPI	223	140M
Equipamentos de Proteção Individual	207	116M
estado de Emergência de Saúde Pública	36	20M
Hospital de Campanha	52	12M
RT-PCR	26	16M
teste rápido	56	35M
ventiladores pulmonares	10	6,6M
ventilador pulmonar	18	12M
Total	2108	950MB

Tabela 1: *Quantidade de arquivos para cada diretório que representa um termo de busca*

Apesar do número de arquivos variar entre as buscas, a proximidade semântica dos termos buscados nos levou à hipótese que poderia haver redundância no conjunto de dados. Ou seja, o mesmo arquivo poderia ter sido recuperado independentemente em duas ou mais buscas.

De fato, esta hipótese está verificada, já que ao analisarmos o conteúdo de todos os arquivos, foi visto que apenas 1011 deles são de fato únicos e, cada um deles, poderia estar presente em pelo menos 1 e até 11 das categorias distintas. Então contamos quantos diretórios cada arquivo está presente. Na tabela 1, a primeira coluna representa a quantidade de diretórios que cada arquivo está e a segunda coluna a quantidade de arquivos que tem essa propriedade

Diretórios com arquivos duplicados	Quantidade de arquivos
1	474
2	242
3	156
4	65
5	45
6	19
7	10
total	1011

Tabela 2: *Quantidade de diretórios que um mesmo arquivo pode pertencer pela quantidade de arquivos.*

2.1 Formatação dos dados

Cada arquivo é uma representação em texto de um documento que originalmente estava em PDF, por conta disso, a formatação dos diários acaba sendo embutida também nos *txt's*. Com isso, cada arquivo que temos em mãos possui uma formatação distinta entre cada um deles e até entre as páginas do mesmo. Alguns possuem o *layout* disposto em colunas, sendo observados de 2 até 4, outros possuem colunas em algumas páginas e outras apenas um corpo centralizado. Existem diários que possuem tabelas dentro do documento.

Durante o processo de conversão dos PDF's para o txt, algumas informações se perderam tais como a quebra de página. O documento de texto é contínuo durante todo seu comprimento, por isso, precisamos recorrer a heurísticas para determinar onde uma página termina e outra começa. O processo descoberto por nós se deu ao abrir os documentos num editor de texto chamado Vim, que mostra um caractere cinza-claro1 justamente na primeira linha de cada página. Este caractere pode ser procurado utilizando expressão regular ou alguma outra ferramenta de busca em textos pelo código `\014`.

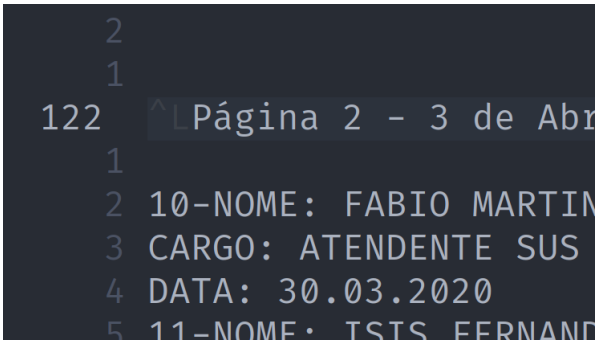
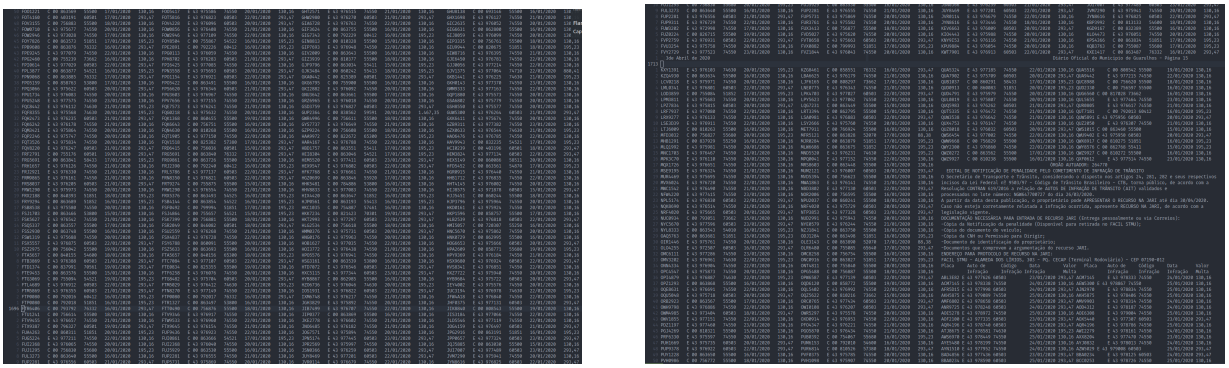


Figura 1: *O primeiro caractere da linha 122, `^L`, aparece em toda primeira linha das segunda página em diante*

Outro problema relacionado com a conversão dos formatos é a manutenção do layout original do PDF, os documentos costumam ter sessões com múltiplas colunas ou tabelas em seu conteúdo. Nada disso sendo padronizado durante a extensão do documento. Como exemplo, podemos verificar o conteúdo do arquivo `122.txt`, que possui algumas páginas com 2 colunas 2, mas também possui tabelas 3 e uma combinação de texto com tabela 3.

Figura 2: *Exemplo de uma página que possui duas colunas*



(a) *Apenas tabela* **(b)** *Tabela com texto*

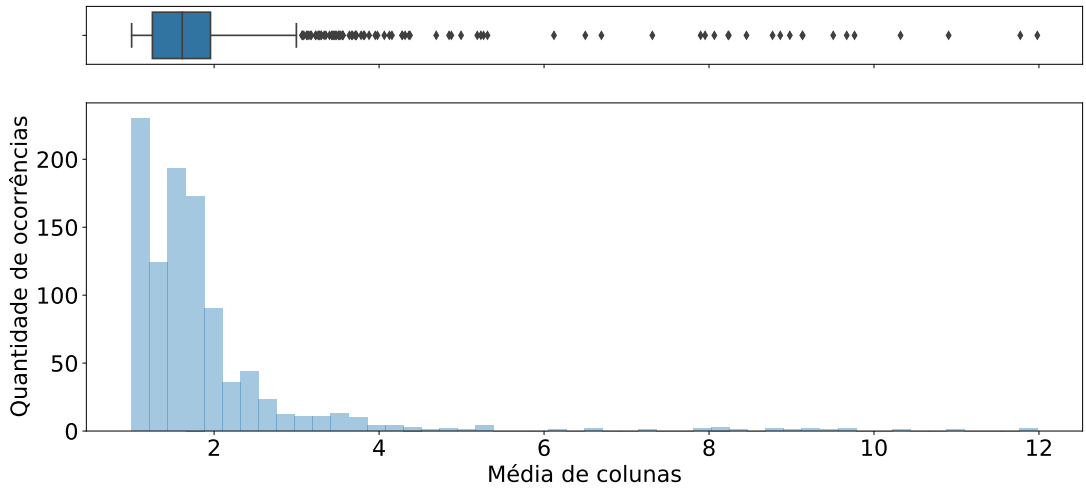
Figura 3: *Formas que tabelas podem aparecer no texto*

Entender esse tipo de detalhe de formatação é importante para nosso objetivo, pois, muitas das análises de texto acabam precisando recuperar informações relacionadas ao contexto das palavras, i.e, outras palavras que estão ao seu redor. Para recuperar termos que estão na vizinhança, temos que transformar as colunas (quando houverem) para um texto sequencial. Apenas com o texto separado a coleta de *n-gramas* se torna possível. Portanto, desenvolver um algoritmo ou heurística que consiga separar as colunas do texto e dispô-las de forma sequencial é imprescindível antes de qualquer análise mais sofisticada. Como não temos conhecimento de nada já pronto que faça o trabalho, decidimos que uma solução autoral seria desenvolvida, mas, para isso, precisamos entender com mais detalhes como este fenômeno ocorre.

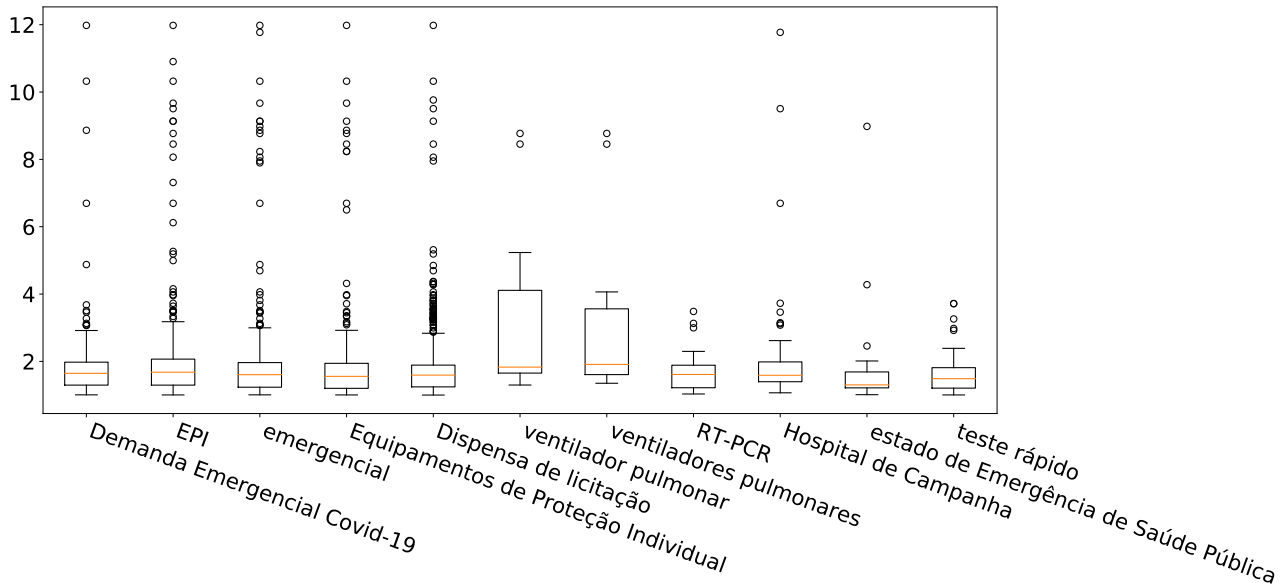
Durante uma análise visual dos textos, percebemos que o espaço entre palavras da mesma coluna era sempre de uma unidade e que na grande maioria das linhas as colunas tinham um espaçamento mínimo de dois espaços. Por conta disso a forma que desenvolvemos para calcular a quantidade de colunas (e seu espaçamento) se dá justamente nesta quase padronização. A análise que decorre do seguinte procedimento: para cada arquivo, separá-lo em páginas (usando o `^L` como delimitador) e depois contar, por página, quantas linhas possuem uma sequência de dois ou mais espaços contíguos. Se a linha possui duas sequencias de espaços, então ela possui três colunas. Da mesma forma, se nenhuma sequência for retornada é porque o texto está em apenas uma coluna. Depois de executar este cálculo para todas as linhas da página, calculamos a média aritmética e resultado é a *média de colunas por página por documento*. O espaçamento entre as colunas é similar; calculamos o comprimento das sequências de espaços e depois vemos a média de espaços, também por página. Chamamos isto de *espaçamento médio entre colunas por página por documento*.

O resultado deste método nos retorna, para cada texto, duas listas de números reais. Cada posição dessa lista representa uma das páginas e o valor a média da quantidade de colunas ou do espaçamento entre as colunas. Com essas listas, fizemos dois tipos agrupamentos diferentes: um contando todos os 1011 arquivos distintos como um único corpo, e outro separando pelas categorias de busca.

Na figura 4 podemos observar duas sub-figuras; a *(a)* representa a quantidade média de colunas considerando todos os documentos e a *(b)* para cada termo de busca. Podemos observar que existem picos perto dos números 1 e 2, indicando que essas são as maiores ocorrências, e, no *boxplot* acima, temos o valor confirmado, com a mediana perto de 1.5. Precisamos levar em conta, portanto, que a maioria das páginas tem 1 ou 2 colunas e poucas tem muito mais do que isso. O maior valor observado foram 13 colunas numa página que continha uma tabela. Agrupando pelos termos de busca observamos uma consistência entre os valores, indicando que não existe quase nenhuma correlação entre a quantidade de colunas e o termo de busca. No caso de *ventiladores pulmonares* e *ventilador pulmonar* observa-se um comportamento um pouco diferente, mas isso se deve pelo desbalanço dos dados, já que segundo a tabela 1, estes são os termos com menos resultados.



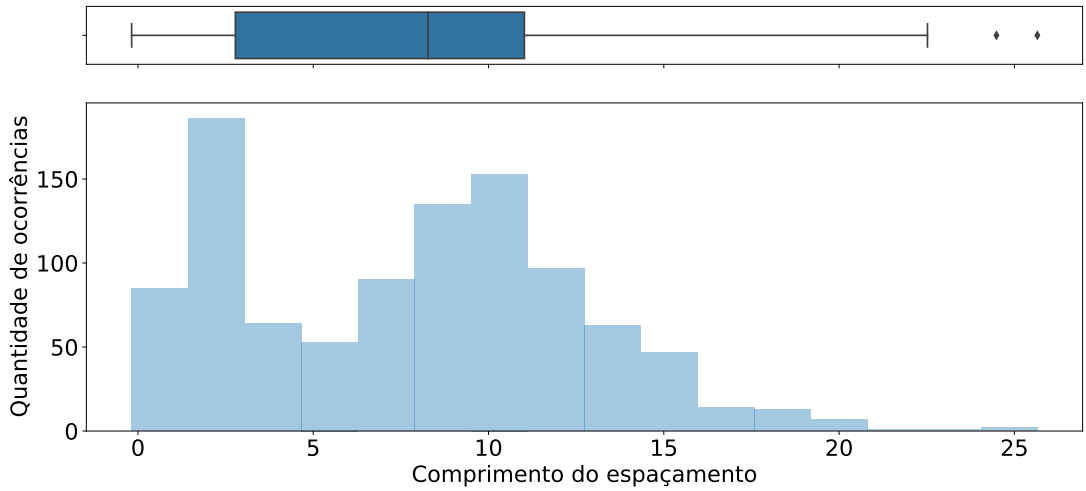
(a) *Quantidade média de colunas para todos os documentos*



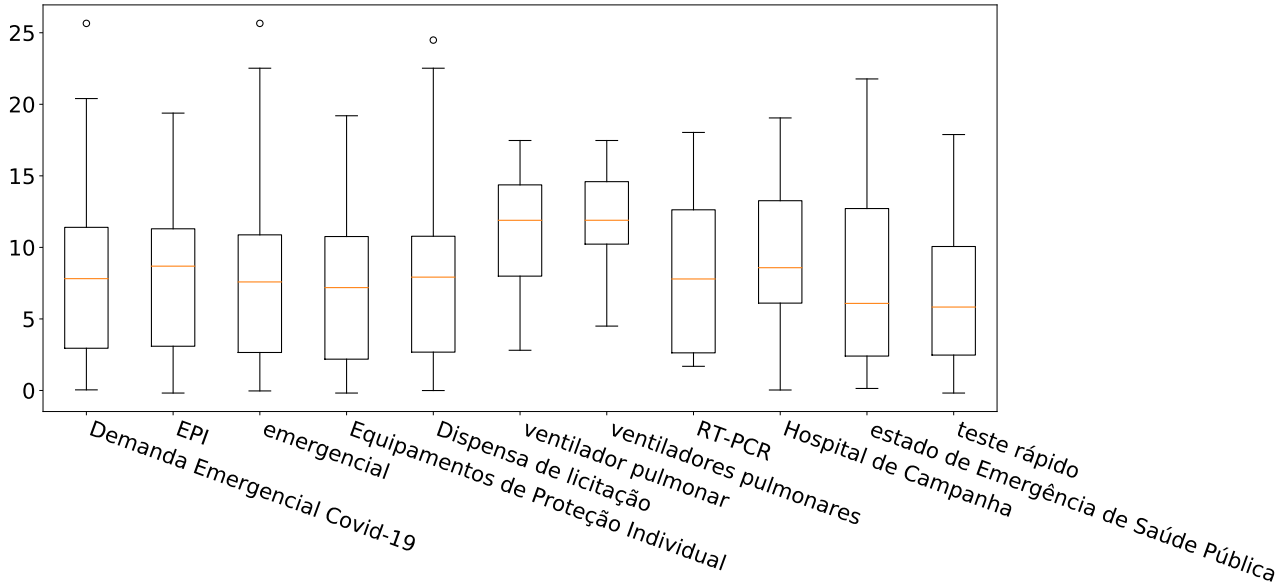
(b) *Quantidade de colunas para cada categoria de busca*

Figura 4: *Análise da quantidade de colunas*

Observando pelo prisma do espaçamento entre as colunas, na figura 5, existem dois picos bem claros por volta de 2,5 espaços e de 11 espaços, sendo o último o mais frequente. Este resultado é consistente com o da quantidade de colunas, já que nele vemos que existe, principalmente, dois casos, um que temos algo em torno de 1 a 3 espaços (provavelmente nas tabelas) e outro com 10 espaços (provavelmente nas colunas que possuem texto). Não foi observado na figura da quantidade de colunas 4 os dois picos, pois as tabelas variam muito na quantidade de colunas, mas não no espaçamento entre elas.



(a) *Espaçamento entre as colunas para todos os documentos*



(b) *Espaçamento entre as colunas para cada categoria de busca*

Figura 5: *Análise do espaçamento entre as colunas*

Em resumo, o que estes dados nos sugerem é que a construção de um procedimento que linearize os documentos é uma tarefa possível, já que algum tipo de comportamento padrão foi observado.

3 Análise de contexto

3.1 Avaliando o entorno dos termos buscados

Nesta seção iremos avaliar o contexto dos termos de busca através de uma inspeção das palavras do entorno do termo utilizado para busca. Apesar dos formatos variados, decidimos aplicar ferramentas básicas de processamento de linguagem natural para obter melhores *insights* sobre os textos.

Para realizar tal tarefa, inicialmente utilizamos a ferramenta *NLTK*, que é uma biblioteca em Python para utilização em atividades de processamento de linguagem natural. Com ela podemos transformar o texto em conjuntos de *Tokens*. Um *Token* é o nome técnico para um sequência de caracteres que queremos tratar como uma unidade, tais como "cabelo"e "violão". Uma ferramenta que nos interessa para esse momento é o "concordance". O "concordance"nos permite analisar os *tokens* do entorno do termo de interesse.

```
.br AVISO DE LICITAÇÃO GABINETE DO PREFEITO ( COM ITEM DE AMPLA PARTICIPAÇÃO E
1410134-35 O Excelentíssimo Senhor Prefeito Municipal de Campinas , usando das
s foram conferidas pelo Exmo . Sr. Prefeito Municipal de Campinas e , de acord
APTO PORTARIA ASSINADA PELO SENHOR PREFEITO PORTARIA N.° 93534/2020 GRAZIELA A
TISTA APTO O Excelentíssimo Senhor Prefeito Municipal de Campinas , usando das
MEIRO APTO O Excelentíssimo Senhor Prefeito Municipal de Campinas , usando das
O SERVIDOR O Excelentíssimo Senhor Prefeito Municipal de Campinas , usando das
ão da Rede Mário Gatti , sito Av . Prefeito da Procuradoria Jurídica ( docs.23
```

(a) *Análise de contexto baseado em tokens do entorno.*

```
                                    prefeito
10134-35       o excelentíssimo senhor prefeito municipal de campinas, usando das atrib
                                    prefeito municipal de campinas e, de acordo
                                    prefeito
                                    prefeito municipal de campinas, usando das atrib
                                    prefeito municipal de campinas, usando das atrib
                                    prefeito municipal de campinas, usando das atrib
de pregão da rede mário gatti, sito av. prefeito
```

(b) *Análise de contexto baseado em caracteres do entorno.*

Figura 6: *Análise do contexto do termo "prefeito"*

Na figura 6a temos a analise do termo "prefeito"utilizando o *NLTK*. O resultado não nos serve para o momento pois a função não captura o problema do formato do arquivo com o qual estamos trabalhando (txt - linha continua) como comentado na seção anterior, além disso, a função não retorna informação para que possamos acumular o resultado em diferentes arquivos para uma análise mais extensiva, apenas a imprime no terminal.

Sendo assim, criamos uma função para análise de entorno do termo de interesse considerando uma quantidade fixa de caracteres antes e após o termo de interesse. Na figura 6b temos um exemplo do resultado de avaliação do mesmo termo de busca no entanto com uma amostra de quantidade fixa de caracteres que o antecedem e o sucedem.

Utilizando a análise do entorno com base na quantidade de caracteres podemos verificar na figura 7 a importância da análise de colunas textuais presente nos documentos. Na primeira linha desta figura podemos perceber que a parte que antecede o termo de busca "teste rápido"se tratava de um texto colunar: ", apresentar a certidão de óbito; se divor- "e após isso o termo buscado e as palavras que fazem sentido ao termo buscado "teste rápido coronavírus (covid19) ...".

```
, apresentar a certidão de óbito; se divor-      - teste rápido coronavírus (covid19) em caráter emergen-
                                                   teste rápido para coronavírus contendo 10 unidades, objetivando a doação
                                                   teste rápido molecular (gene xpert), com fornecimento e
                                                   teste rápido medteste             atribuições legais conferidas pelo at
parágrafo primeiro - as farmácias que oferecerem o teste rápido deverão notificar
                                                   teste rápido utilizado, os valores de
                                                   teste rápido distribuído pelo ministério da saúde. este estudo, divulgad
parágrafo primeiro - as farmácias que oferecerem o teste rápido deverão notificar
```

Figura 7: *Análise do contexto do termo "prefeito"*

Esta avaliação poderá nos auxiliar na busca de diferentes contextos para um termo de busca, identificar padrões que possam nos auxiliar nas implementações e como comentado, a importância da transformação de textos que se apresentam em colunas para um formato sequencial.

3.2 Análise de palavras

Após a *tokenização*, para separar as sentenças em palavras que podemos obter uma informação que facilite o processamento e seu entendimento, dois filtros foram feitos, o primeiro retirando as *stopwords*. *Stopwords* são palavras comuns que pouco contribuem para o significado de uma frase. O *NLTK* fornece um lista de *stopwords* na língua portuguesa.

Além disso, também incluímos algumas palavras específicas do contexto ao qual os diários oficiais são escrito. Após isso, retiramos palavras que tem menos de 4 caracteres, visto que após uma observação, percebemos que estas palavras, na maioria dos casos, eram pedaços de outras palavras resultante da quebra de linhas do texto.

Criamos um gráfico de barras que mostra as 30 palavras mais frequentes de todos os textos após serem filtrados os termos que não nos interessava.

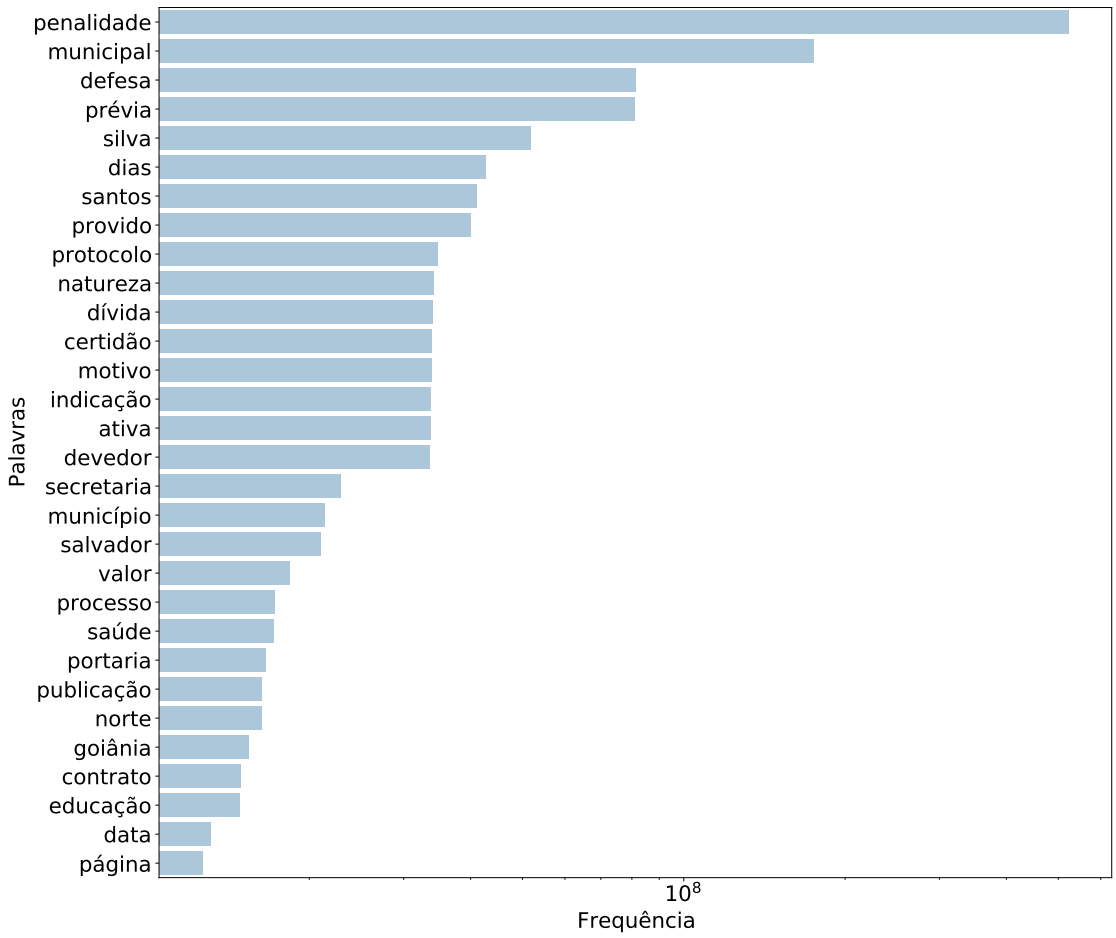


Figura 8: *Frequência de palavras*

As palavras apresentadas no gráfico 8 aparecem muitas vezes nos nossos textos. Acreditamos que algumas delas possuem bastante relevância na extração de informações dos diários e que possam auxiliar na exploração e compreensão das medidas adotadas por municípios para enfrentar a pandemia do novo coronavírus.

4 Conclusão

O objetivo deste trabalho foi entender o processo: desde a origem dos dados que iremos trabalhar; passando pela coleta feita pelos membros da OKF, no agrupamento por palavras-chave; terminando no tipo de problema que eles pretendem solucionar. Este trabalho foi importante para percebemos que para a implementação de um *framework* de cruzamento e levantamento de dados contra a corrupção, precisávamos entender mais a fundo o que de fato tínhamos em mãos. Para isso, tomamos uma abordagem partindo do dado concreto culminando para uma informação abstrata: a linguagem em si.

Verificamos como os documentos estão dispostos em diretórios e qual o tamanho de cada um. Identificamos presença de arquivos duplicados nos dados. Isso é importante conhecer pois pode influenciar bastante o processamento. Após esta etapa, realizamos o processo de analisar o conteúdo dos arquivos, mais especificamente no que tange a formatação. Percebemos que é bastante comum a presença de texto em múltiplas colunas, e presença de dados em tabelas diversas, sem nenhum tipo de padrão obvio. Ao contar a quantidade de colunas por página, vimos que existe alguma padronização e que, por isso, e isso nos instiga a investigar se é possível executar algum procedimento que separe esse texto caso venha a ser necessário. Por último, realizamos uma

análise semântica. Esta se fez necessária para que pudéssemos entender o contexto que cada documento esta inserido. Isso nos permite, por exemplo, conhecer quais jargões são utilizados. Outra análise realizada foi a sobre as palavras que ocorriam nas redondezas das palavras-chave que foram utilizadas para a obtenção dos arquivos.