Recuperando dados com o ElasticSearch

Gustavo Salvador Soares,Victor Pedro Rodrigues Lisboa

1

1. O que é o ElasticSearch

O ElasticSearch é uma ferramenta de busca baseada no Apache Lucene, onde múltiplos clientes podem realizar buscas de texto completo em uma interface web e orientada à JSON.

1.1. Guia de instalação

1.1.1. Em ambientes com sistema operacional baseado no debian

É possível instalar de duas formas diferentes, uma delas é utilizando o gerenciador de pacotes e a outra é instalando manualmente

Para instalar utilizando o gerenciador de pacotes deve-se executar os seguintes passos:

1. talvez seja necessário instalar o transport-https

```
sudo apt-get install apt-transport-https
```

2. é necessário adicionar o repositório à lista de repositórios do gerenciador de pacotes

```
echo "deb␣https://artifacts.elastic.co/packages/7.x/
   apt␣stable␣main" | sudo tee -a /etc/apt/sources.
   list.d/elastic-7.x.list
```

3. instale o elasctisearch utilizando o gerenciador de pacotes

```
apt-get update && sudo apt-get install elasticsearch
```

Para instalar manualmente deve-se executar os seguintes passos:

1. Baixar o arquivo .deb diretamente do repositório

```
wget https://artifacts.elastic.co/downloads/
   elasticsearch/elasticsearch-7.0.1-amd64.deb
```

2. Baixar o checksum para verificar a integridade dos arquivos

```
wget https://artifacts.elastic.co/downloads/
   elasticsearch/elasticsearch-7.0.1-amd64.deb.sha512
```

3. verificar a integridade do arquivo baixado realizando o checksum

```
shasum -a 512 -c elasticsearch-7.0.1-amd64.deb.sha512
```

4. instalar o pacote manualmente

```
sudo dpkg -i elasticsearch-7.0.1-amd64.deb
```

1.1.2. Em ambientes com sistema operacional baseado no RPM, tais como fedora, OpenSuSe, CentOS,Red Hat e Oracle

É possível instalar de duas formas diferentes, uma delas é utilizando o gerenciador de pacotes e a outra é instalando manualmente

Para instalar utilizando o gerenciador de pacotes deve-se executar os seguintes passos:

1. crie um arquivo chamado elasticsearch.repo em /etc/yum.repos.d(para distribuições Red Hat) ou /etc/zypp/repos.d/ (para distribuições OpenSuSe) contendo:

```
[elasticsearch-7.x]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-
   elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

2. instale o elasctisearch utilizando o gerenciador de pacotes

```
sudo yum install elasticsearch (CentOS, Red Hat)
sudo dnf install elasticsearch (Fedora, Red Hat)
sudo zypper install elasticsearch (OpenSuSe)
```

Para instalar manualmente deve-se executar os seguintes passos:

1. Baixar o arquivo .deb diretamente do repositório

```
wget https://artifacts.elastic.co/downloads/
   elasticsearch/elasticsearch-7.0.1-x86_64.rpm
```

2. Baixar o checksum para verificar a integridade dos arquivos

```
wget https://artifacts.elastic.co/downloads/
   elasticsearch/elasticsearch-7.0.1-x86_64.rpm.sha512
```

3. verificar a integridade do arquivo baixado realizando o checksum

```
shasum -a 512 -c elasticsearch-7.0.1-amd64.prm.sha512
```

4. instalar o pacote manualmente

```
sudo rpm --install elasticsearch\-7.0.1\-x86_64.rpm
```

1.1.3. Em ambientes com sistema operacional Windows

Para instalar o elasticSearch em ambientes Windows siga os seguintes passos:

1. Baixe o arquivo .msi de https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.0.1.msi
2. Execute o arquivo .msi para abrir a janela de instalação
3. Na primeira tela, de escolher o diretório, escolha o padrão
4. Na próxima tela é necessário escolher como instalar o elasticSearch, como um serviço ou não, se ele for instalado como um serviço, ele será executado no startup
5. A próxima etapa é a de configuração, nela o usuário pode colocar as configurações desejadas no elasticSearch

1.1.4. Instalando utilizando o docker

O elasticSearch também pode ser instalado usando o docker, para isso, siga os seguintes passos, lembrando que todos os comandos do docker devem ser executados com o usuário de root, caso o usuário normal não esteja no grupo do docker:

1. Primeiro é necessário baixar a imagem do repositório

```
docker pull docker.elastic.co/elasticsearch/
   elasticsearch:7.0.1
```

2. Rode o elasticSearch

```
docker run -d -p 9200:9200 -p 9300:9300 -e "discovery.
   type=single-node" docker.elastic.co/elasticsearch/
   elasticsearch:7.0.1
```

2. Utilizando o elasticsearch

O elasticsearch utiliza o protocolo HTTP em conjunto com JSON no body das requisições para realizar todas as suas operações, sejam elas de inserção de dados, atualização de dados, remoção de dados, configuração de recursoos e buscas. Tais operações são realizadas utilizando os métodos de GET,POST,PUT e DELETE.

2.1. Primeiras operações no elasticSearch

Após a instalação e execução do elasticSearch, ele já está pronto para ser utilizado, ele fica executando na porta 9200 do computador, para verificar seu status basta acessar localhost:9200 e verificar se o elasticSearch está rodando, se ele estiver, será retornado um json parecido com o que aparece abaixo.

```
{
  "name" : "jyp2Nwz",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "_na_",
  "version" : {
    "number" : "5.6.16",
```

```
    "build_hash" : "3a740d1",
    "build_date" : "2019-03-13T15:33:36.565Z",
    "build_snapshot" : false,
    "lucene_version" : "6.6.1"
  },
  "tagline" : "You Know, for Search"
}
```

2.1.1. Adicionando um documento

Agora que temos o elasticSearch funcionando precisamos adicionar um documento para realizar as operações de busca mais adiante.Para isso, utiliza-se um método POST, a requisição é feita para a seguinte url localhost:9200/indice/fragmento/idDocumento, e no body da requisição passaremos um JSON, como o que se encontra abaixo.

```
{
   "DOCID": "FSP940101-002",
   "DATE": "940101",
   "DOCNO": "FSP940101-002",
   "TEXT": "Os quenianos dominaram a corrida de So
      Silvestre ontem."
}
```

Após a execução da operação, será retornado um JSON informando se a operação ocorreu com sucesso.

2.1.2. Realizando uma busca

Para realizar uma busca, utilizaremos um método GET, a url para realização da consulta é um pouco diferente, adicionamos _search? ao final, o que indica que estamos realizando uma consulta, logo a url fica localhost:9200/indice/fragmento/_search?q=termoASerBuscado. Também podemos forçar a busca a ser realizada somente em um campo, utilizando o JSON adicionado anteriormente, localhost:9200/indice/fragmento/_search?q=TEXT:quenianos

A busca também pode ser realizada utilizando um JSON no body da requisição, para isso, utilizamos a seguinte url, localhost:9200/indice/fragmento/_seach? e um JSON no body, como por exemplo o que se encontra logo abaixo.

```
{
   "query": {
      "match" : {
         "query" : "silvestre",
         "fields": ["DATE","TEXT"],
         "fuzziness": "AUTO"
      }
   },
```

```
    "_source": ["DOCID","DOCNO","DATE","TEXT"],
    "size": 100
}
```

O campo query é referente à consulta que será realizada, fields mostra os campos do JSON em que o conteúdo de query será procurado, fuzziness é um parametro que é utilizado para tratar similiaridade das palavras, isso é utilizado para prevenir que resultados não apareçam caso ocorra um typo, por exemplo, caso fosse escrito silveste em vez de silvestre, sem o fuziness a consulta não retornaria o documento esperado, size indica a quantidade máxima de documentos que serão recuperados e _source os campos do json que serão retornados.

2.2. O que a ferramenta oferece

O elasticSearch oferece diferentes recursos, desde a parte de indexação e busca de dados, agregação e geração de métricas,análise e tratamento dos dados, utilizando stemming, eleminação de stopwords, sinônimos e diferentes outros recursos que serão mencionados mais adiante.

Além disso, a ferramenta possui diferentes tipos de consultas, como a consulta booleana, boosting query, dis max query

2.2.1. Stemmers

O elasticSearch tem suporte de stemming para diferentes idiomas e dentro de cada idioma existe uma ou mais opções de stemmers, como por exemplo o idioma inglês, possui o porter_stem, kstem, EnglishMinimalStemmer, porter e porter2, ambos usando SnowBall. Para adicionar uma configuração de stemmer, é necessário fazer uma requisição PUT para a seguinte url localhost:9200/indice e passar o JSON com as configurações do stemmer, abaixo segue um exemplo.

```
{
   "index" : {
      "analysis" : {
         "analyzer" : {
            "my_analyzer" : {
               "tokenizer" : "standard",
               "filter" : ["standard", "lowercase", "
                  my_stemmer"]
            }
         },
         "filter" : {
            "my_stemmer" : {
               "type" : "stemmer",
               "name" : "brazilian"
            }
         }
      }
```

```
    }
}
```

2.2.2. Eliminação de stopwords

A ferramenta também possui suporte à eliminação de stopwords de diferentes idiomas, a forma de adicionar essa configuração é a mesma de adicionar um stemmer, através de uma requisição PUT para localhost:9200/indice com o JSON de configuração, é possível definir uma lista de stopWords ou utilizar a lista da ferramenta. Abaixo segue um exemplo utilizando uma lista de stopwords

```
{
   "settings": {
      "analysis": {
         "filter": {
            "my_stop": {
               "type": "stop",
               "stopwords": ["a", "de", "para"]
            }
         }
      }
   }
}
```

Agora um exemplo utilizando a lista já pré-definida da plataforma

```
{
   "settings": {
      "analysis": {
         "filter": {
            "my_stop": {
               "type": "stop",
               "stopwords": "_brazilian_"
            }
         }
      }
   }
}
```

2.2.3. Sinônimos

Também é possível tratar sinônimos durante o processo de análise. Os sinônimos são configurados utilizando um arquivo de configuração e uma lista de sinônimos. No exemplo utilizares um arquivo chamado synonym.txt que contém uma lista de sinônimos que será lida pela ferramenta. Tal configuração é aplicada através de uma requisição PUT para localhost:9200/indice passando um JSON no body. Abaixo um exemplo de configuração

```
{
   "settings": {
      "index" : {
         "analysis" : {
            "analyzer" : {
               "synonym" : {
                  "tokenizer" : "whitespace",
                  "filter" : ["synonym"]
               }
            },
            "filter" : {
               "synonym" : {
                  "type" : "synonym",
                  "synonyms_path" : "analysis/synonym.txt"
               }
            }
         }
      }
   }
}
```

2.2.4. Relevância de campos

No elasticSearch era possível aumentar a relevância de campos de duas formas diferentes, uma delas era utilizando um arquivo de configuração, no qual que deve ser carregado da mesma maneira que os outros, só que essa função ficou deprecated na versão 5.0.0, atualmente esse boost do score só poed ser feito durante a consulta. Essa consulta aceita duas sub consultas, uma positiva e uma negativa, a lista de resultados conterá somente os resultados encontrados na sub consulta positiva mas os documentos que estiverem na positiva e também estiverem na negativa, terão seu score original alterado, ele será multiplicado pelo valor setado no parâmetro negative_boost. Abaixo um exemplo de umas consulta utilizando essa ideia.

```
{
   "query": {
      "boosting" : {
         "positive" : {
            "term" : {
               "campo1" : "valor"
            }
         },
         "negative" : {
             "term" : {
                "campo2" : "valor2"
            }
```

```
            },
            "negative_boost" : 0.2
        }
    }
}
```

2.3. Como o elasticSearch calcula o ranking

O elasticSearch utiliza a Lucene's Pratical Score Function para calcular o score dos documentos, tal método utiliza consultas booleanas, o método TF/IDF e o Modelo Vetorial e combina os três para calcular o valor do score. As queries realizadas no elasticSearch nem sempre são booleanas, nesse trabalho só utilizamos consultas match por exemplo, nesses casos a própria ferramenta faz uma conversão para query booleana para poder calcular o score, abaixo podemos observar um exemplo dessa conversão.

Listing 1. consulta match

```
{
  "query": {
    "match": {
      "text": "quick fox"
    }
  }
}
```

Listing 2. consulta convertida para booleana

```
{
  "query": {
    "bool": {
      "should": [
        {"term": { "text": "quick" }},
        {"term": { "text": "fox" }}
      ]
    }
  }
}
```

A consulta booleana implementa o modelo booleano, assim que um resultado é encontrado o score é calculado, combinando o score de cada termo.A fórmula utilizada é chamada Pratical Scoring Function. Abaixo temos a fórmula utilizada.

```
score(q,d) = (1)
          queryNorm(q) (2)
         coord(q,d) (3)
          ( (4)
             tf(t in d) (5)
            idf(t) (6)
            t.getBoost() (7)
```

```
            norm(t,d) (8)
        ) (t in q) (9)
```

Agora vamos analisar a fórmula por partes,

1. (1) é o score do documento d para a consulta q
2. (2) a query é normalizada, para que os resultados de uma possam ser comparados com o resultado de outra. Esse fator é calculado utilizando a seguinte fórmula, queryNorm = 1 / sumOfSquaredWeights , onde sumOfSquaredWeights é calculado somando os valores do IDF dos termos da query elevado ao quadrado.
3. (3) chamado de coordination factor, é uma forma de recompensar os documentos que possuem mais termos da consulta. o score é multiplicado pelo número de termos da consulta que aparecem no documento e dividido pelo total de termos na query.
4. (4) e (9) somatório dos pesos de um termo t da consulta q no documento d
5. (5) é a frequencia do termo t no documento d
6. (6) é o valor de IDF do termo t
7. (7) é o boost que foi aplicado a consulta
8. (8) é uma normalização aplicada no boost de um campo, é calculada fazendo a raiz quadrada da quantidade de termos no campo.

3. Execução do Trabalho

Para começar a executar o trabalho, além de baixar e instalar o elasticSearch também foi necessário buscar uma forma de se comunicar com ele, via interface gráfica ou linha de comando ou API. Depois de descobrir como se comunicar com a ferramenta foi necessária saber como tratar o texto para tentar se obter os melhores resultados possíveis, para isso além de elimininação de stopwords foram utilizados diferentes experimentos, que serão melhor explicados abaixo.

3.1. Como é feito o acesso à ferramenta

A ferramenta trabalha com requisições HTTP utilizando os métodos GET, POST, PUT e DELETE, por isso ela permite que a comunicação seja feita de diferentes formas, nesse trabalho vamos falar um pouco mais sobre a comunicação utilizando a API Java do elasticSearch e utilizando o Postman.

3.1.1. API Java

O elasticSearch possui uma API em java que permite que uma aplicação se conecte à ferramenta e possa executar as operações na plataforma. Para isso é necessário que o projeto tenha o elasticSearch como dependência, no nosso caso foi utilizado o maven como gerenciador de dependências, abaixo se encontra o código que deve ser adicionado ao arquivo pom.xml (arquivo que contém as dependências e configurações de um projeto maven

```
<dependency>
      <groupId>org.elasticsearch.client</groupId>
      <artifactId>transport</artifactId>
```

```
      <version>5.6.16</version>
</dependency>
```

Além disso, também utilizamos a dependência do JSON, que não faz parte do pacote básico da JDK. Abaixo segue o código para adicionar a dependência ao projeto.

```
<dependency>
      <groupId>org.json</groupId>
      <artifactId>json</artifactId>
      <version>20180130</version>
</dependency>
```

Agora com as dependências ja adicionadas ao projeto é necessário se comunicar com o cliente do elasticSearch que está sendo executado. No projeto foi criada uma classe chamada ElasticSearchAction que é responsável por realizar todas as operações no elasticSearch, inclusão de documento, exclusão, atualização e consulta. Abaixo segue exemplo de código que adiciona um documento na ferramenta.

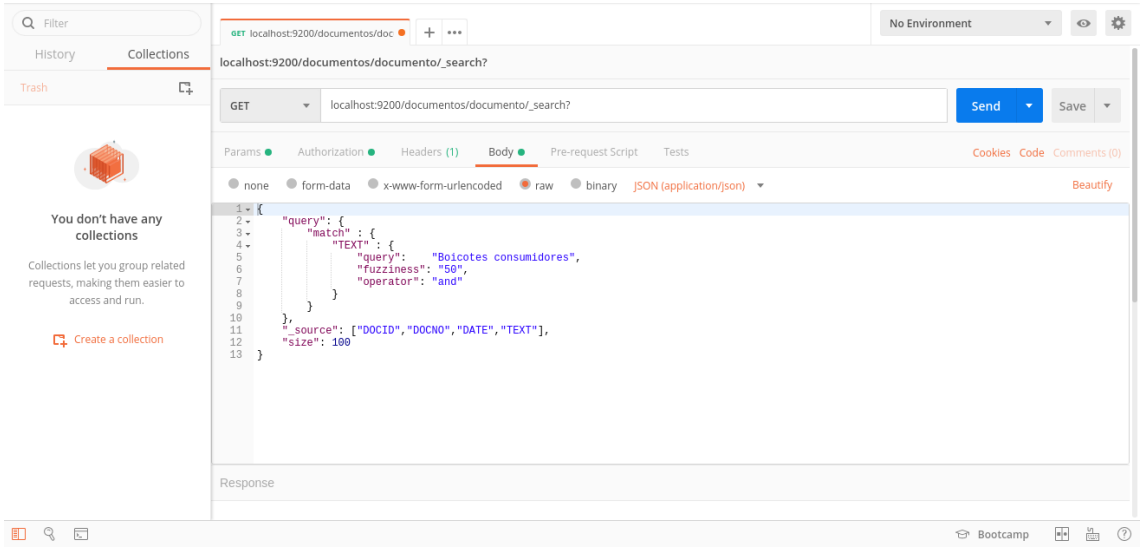
```
   Settings settings = Settings.builder().put("cluster.name
      ", "docker-cluster").build();

   TransportClient client = new PreBuiltTransportClient(
      settings)
                              .addTransportAddress(new
                                 InetSocketTransportAddress
                                 (InetAddress.getByName("
                                 localhost"), 9300));

   client.prepareIndex("documentos", "documento", 1)
                              .setSource(jsonBuilder().
                                 startObject().field("DOCID
                                 ", "19854596")
                                          .field("DOCNO", "
                                             4758435345")
                                          .field("DATE", "
                                             20011995")
                                          .field("TEXT", "
                                             Esse␣␣um␣
                                             exemplo␣de␣
                                             como␣adicionar
                                             ␣um␣indice␣em␣
                                             JAVA.").
                                             endObject())
                              .get();
```

A chamada ao método prepareIndex recebe como parâmetros o nome do íncide, o nome do fragmento e o id do documento que está sendo adicionado.A chamada ao setSource indica o objeto que será adicionado, dentro da chamada deste método é feita

uma chamada ao jsonBuilder, que é responsável por montar o JSON que será passado ao cliente para ser adicionado.

3.1.2. Postman

Além da API em Java foi utilizado o Postman, ele é uma ferramenta que permite que enviar e receber respostas de uma API, é uma ferramenta aberta, que pode ser baixada através do link https://www.getpostman.com/downloads/ e executada, sem necessidade de instalação. A interface do Postman é bem amigável, com isso, faz com que seja fácil se conectar à ferramenta e realizar as requisições. O postman foi mais utilizado na hora dos experimentos, uma vez que o tratamento dos dados foi feito utilizando uma aplicação JAVA, popular os índices utilizando a API java foi mais fácil. Abaixo segue uma imagem da tela do postman



Na figura conseguimos observar as abas Authorization, que deve ser utilizada quando a API possui uma autenticação, Body, que foi a única utilizada, pois nela colocamos o JSON utilizado na busca. Na barra de endereço colocamos a url da API com o índice e fragmento que desejamos utilizar na busca, além de selecionar o método HTTP desejado.

3.2. Como os documentos foram tratados

Antes dos documentos serem inseridos, eles foram formatados e tratados. Os arquivos com as consultas foram lidos por uma aplicação Java e parseados, de forma que cada documento foi transformado em uma lista. Na lista foi utilizado um Regex para pegar o conteúdo entre as tags e esse conteúdo foi utilizado para popular um JSON, para assim pode ser adicionado ao elasticSearch. Porém, o texto do documento, antes de ser adicionado ao JSON, ele foi passou por um método que foi responsável por eliminar as stopWords, não utilizamos stemmer porque estávamos utlizando o fuzziness nas consultas, o que nos permitiu obter hits de palavras no plural quando utilizávamos a forma da palavra no singular.

3.3. Como as consultas foram tratadas

O tratamento das consultas foi mais simples, o arquivo de consultas foi lido e então foi utilizado um regex para pegar apenas o título das perguntas. Após termos o título utilizamos uma eliminação de stopwords com o objetivo de obter um melhor score. Após eliminar as stopwords, já possuíamos as palavras chave que seriam utilizadas nas buscas.

4. Experimentos Realizados

Na coleção de testes foram realizados alguns experimentos para analisarmos como o score era alterado e ver formas de tentar melhorar o mesmo.As consultas foram realizadas utilizando o multi_match, que permite que eu passe mais de um campo para as consultas, no caso de usar somente um campo, utilizar o match teria o mesmo efeito.

4.1. Experimento do Fuzziness

Foi realizado uma vez a consulta utilizando o operador Fuzziness e uma vez sem o operador, a fim de observar o impacto do fuzziness no score. Abaixo segue as consultas e os resultados encontrados.

Listing 3. Consulta

```
{
   "query": {
      "multi_match" : {
         "query" : "Boicotes consumidores",
         "fields": ["DATE","TEXT"],
         "fuzziness": "AUTO"
      }
   },
   "_source": ["DOCID","DOCNO","DATE","TEXT"],
   "size": 5
}
```

Listing 4. Resultado com Fuzziness

```
{
   "_index": "documentos",
   "_type": "documento",
   "_id": "87635",
   "_score": 22.856691,
   "_source": {
      "DOCID": "FSP951002-033",
      "DATE": "951002",
      "DOCNO": "FSP951002-033",
      "TEXT": "[especial, folha, lisboa, resultado, oficial
         , eleies, portuguesas, s, sair, 15, dias,
         responsabilidade, , pequenas, sees, decidiram,
         boicotar, eleies, boicotes, esto, ligados, questes
         , prticas, gua, esgoto, asfalto, escolas, regies,
         lei, portuguesa, prev, caso, sees, no, consigam,
```

```
         promover, votao, tentativas, aps, prazo, dias,
         total, 15, sees, eleitorais, reunindo, cerca, 15,
         eleitores, boicotaram, eleies, boicotes, ganharam,
          fora, eleies, 1991, psd, conquistou, deputado,
         lisboa, vila, santa, maria, decidiu, no, abrir,
         seo, eleitoral, gua, encanada, seo, santa, maria,
         300, eleitores, faltavam, 190, votos, partido,
         socialista, revolucionrio, eleger, deputado, psd,
         conseguiu, manter, boicote, chegou, gua, encanada,
          boicotes, acabaram, legitimados, motivos, boicote
         , vo, mau, cheiro, fbrica, peixe, so, romo,
         coronado, porto, (norte), excesso, buracos,
         estrada, longriva, bragana, (norte), alm, gua,
         benafim, sul, pas, (jr)]"
   }
},
{
   "_index": "documentos",
   "_type": "documento",
   "_id": "16635",
   "_score": 14.960273,
   "_source": {
      "DOCID": "FSP950803-044",
      "DATE": "950803",
      "DOCNO": "FSP950803-044",
      "TEXT": "[vinicius, torres, freire, paris, governo,
         australiano, convocou, estados, compem, pas,
         boicotar, empresas, francesas, medida, , represlia
         , retomada, testes, nucleares, frana, boicotes,
         australianos, abriram, crise, diplomtica, pases,
         frana, retirou, indeterminado, embaixador,
         austrlia, setembro, frana, explodir, bombas, atol,
          mururoa, (polinsia, francesa, oceano, pacfico),
         ministro, defesa, australiano, robert, ray, pediu,
          edf, (eletricidade, frana), proibida, participar,
          privatizao, us$, 9, bilhes, contrato, compagnie,
         lyonnaise, des, eaux, (de, abastecimento, gua),
         tambm, est, ameaado, anteontem, ray, excluiu,
         dassault, fabricante, francesa, avies, concorrncia
         , us$, 400, milhes, jornal, francs, ``le, monde\",
          dassault, no, chance, alguma, vencer, disputa,
         venda, avies, treinamento, reunio, frum, segurana,
          sia-pacfico, asean, (associao, pases, sudeste,
         asitico), encerrada, emitiu, declarao, pedia,
         testes, nucleares, declarao, no, mencionou, china,
          frana, pases, prosseguem, experincias, atmicas,
```

```
          primeiro-ministro, francs, alain, jupp, disse,
          entrevista, coletiva, governo, continua, firme,
          propsito, realizar, testes, seguida, franois,
          barouin, porta-voz, governo, afirmou, ``ofensiva,
          discriminatria\", australiana, aproveita, motivo,
          testes, nucleares, expandir, influncia, oceania,
          sia, ``a, austrlia, , movida, ambies, econmicas,
          geopolticas, regio, frana, forte, presena\", disse
          , barouin, jornal, ``le, parisien\", divulgou,
          pesquisa, 60%, franceses, consideram, chirac,
          desistir, testes, jornal, ``le, quotidien, paris
          \", traz, pgina, convocao, boicote, produtos,
          pases, estariam, discriminando, frana, (austrlia,
          dinamarca, japo, noruega, zelndia), cadeia, hotis,
           sueca, anunciou, no, servir, vinhos, franceses,
          restaurantes, parlamento, governo, suecos, tambm,
          j, haviam, adotado, medida, agncias,
          internacionais]"
   }
}
```


Listing 5. Resultado sem Fuzziness


```
{
   "_index": "documentos",
   "_type": "documento",
   "_id": "87635",
   "_score": 13.766897,
   "_source": {
      "DOCID": "FSP951002-033",
      "DATE": "951002",
      "DOCNO": "FSP951002-033",
      "TEXT": "[especial, folha, lisboa, resultado, oficial
         , eleies, portuguesas, s, sair, 15, dias,
         responsabilidade, , pequenas, sees, decidiram,
         boicotar, eleies, boicotes, esto, ligados, questes
         , prticas, gua, esgoto, asfalto, escolas, regies,
         lei, portuguesa, prev, caso, sees, no, consigam,
         promover, votao, tentativas, aps, prazo, dias,
         total, 15, sees, eleitorais, reunindo, cerca, 15,
         eleitores, boicotaram, eleies, boicotes, ganharam,
          fora, eleies, 1991, psd, conquistou, deputado,
         lisboa, vila, santa, maria, decidiu, no, abrir,
         seo, eleitoral, gua, encanada, seo, santa, maria,
         300, eleitores, faltavam, 190, votos, partido,
         socialista, revolucionrio, eleger, deputado, psd,
         conseguiu, manter, boicote, chegou, gua, encanada,
          boicotes, acabaram, legitimados, motivos, boicote
```

```
          , vo, mau, cheiro, fbrica, peixe, so, romo,
          coronado, porto, (norte), excesso, buracos,
          estrada, longriva, bragana, (norte), alm, gua,
          benafim, sul, pas, (jr)]"
   }
},
{
   "_index": "documentos",
   "_type": "documento",
   "_id": "32899",
   "_score": 8.678528,
   "_source": {
      "DOCID": "FSP940714-103",
      "DATE": "940714",
      "DOCNO": "FSP940714-103",
      "TEXT": "[surfista, empresrio, roberto, valrio, 34,
         morreu, ltimo, sbado, vtima, aneurisma, cerebral,
         forte, crise, asma, saiu, fbrica, cyclone,
         assistir, jogo, brasil, passou, casa, conseguiu,
         chegar, sozinho, hospital, lagoa, morreu, seguida,
          surfe, perde, competidor, empresrio, patrocinador
         , amigo, *na, ltima, edio, revista, japonesa, \"
         surfing, life\", brasil, brasileiros, ocupam,
         pginas, comeo, mundial, amador, rio, time,
         brasileiro, wct, seo, fixa, fotgrafo, alberto,
         sodr, publicidade, empresas, nacionais, fabio,
         gouveia, conquistando, bicampeonato, ocean, dome,
         \u0096a, piscina, ondas\u0096, promovendo, filme,
         \"hawaii, nine, 4, \u0096, the, gathering\",
         ratificam, importncia, brasileira, cenrio, mundial
         , *comeou, triagem, 26, etapa, wqs, gunston, 500,
         principal, evento, temporada, estrelas, distribuir
         , us$, 80, prmios, 2, 500, colocado, boicotes,
         graas, apartheid, campeonato, termina, domingo,
         durban, frica, sul, sequncia, rola, etapa, wqs,
         billabong, country, feeling, jeffrey's, bay, *
         comea, prxima, quarta-feira, rio, surf, beach,
         show, reunindo, principais, empresas, surfwear,
         pas, evento, acontece, riocentro, rio]"
   }
}
```


Conseguimos observar que com o fuzziness os scores foram quase o dobro do que sem o fuzziness. Outro resultado observado é que a segunda consulta retornada foi diferente, o segundo resultado da consulta sem o fuzziness trouxe um resultado que faz menos sentido aparecer em segundo do que o resultado em segundo na consulta com o fuzziness.

4.2. Experimento utilizando o Operator

Foi realizado a mesa consulta do documento anterior só que dessa vez foi utilizado o operator AND, ele faz com que só retornem consultas que possuem todos os termos utilizados na busca.

Listing 6. Consulta

```
{
   "query": {
      "multi_match" : {
         "query" : "Boicotes consumidores",
         "fields": ["DATE","TEXT"],
         "fuzziness": "AUTO",
         "operator" : "and"

      }
   },
   "_source": ["DOCID","DOCNO","DATE","TEXT"],
   "size": 5
}
```

Listing 7. Resultado

```
{
   "_index": "documentos",
   "_type": "documento",
   "_id": "86574",
   "_score": 13.213015,
   "_source": {
      "DOCID": "FSP940802-036",
      "DATE": "940802",
      "DOCNO": "FSP940802-036",
      "TEXT": "[agncia, folha, belo, horizontecerca, 40,
         donas-de-casa, belo, horizonte, saram, ruas,
         cidade, protestar, frente, principais,
         supermercados, aumentos, abusivos, preos,
         manifestao, organizada, movimento, donas-de-casa,
         minas, alm, comemorar, ms, lanamento, real,
         aproveitou, decretar, nacional, boicote, consistiu
         , tentativa, convencer, consumidores, no, comprar,
          produtos, preos, considerados, abusivos,
         presidente, movimento, lcia, pacfico, homem, disse
         , manifestao, \"positiva\", pm, no, registrou,
         incidentes, manh, trs, principais, hipermercados,
         cidade, \u0096bon, march, paes, mendona, extra\
         u0096, fecharam, portas, at, 13h, presidente,
         associao, mineira, supermercados, jos, nogueira,
         disse, fechamento, no, passou, \"coincidncia\",
         normalmente, 1, ms, estabelecimentos, fecham,
```

```
         portas, balanos, mensais, \"o, boicote,
         incentivado, supermercados, tambm, esto,
         boicotando, fornecedores, insistem, colocar, preos
         , acima, normal\", disse]"
   }
},
{
   "_index": "documentos",
   "_type": "documento",
   "_id": "19869",
   "_score": 13.1532955,
   "_source": {
      "DOCID": "FSP940419-044",
      "DATE": "940419",
      "DOCNO": "FSP940419-044",
      "TEXT": "[indstrias, reajustam, at, 16, 5%, preo,
         produto, passa, variar, semanalmente, maioda,
         reportagem, localo, boicote, , cerveja, brahma,
         iniciado, semana, passada, supermercadistas, so,
         paulo, continua, produto, fica, 16, 5%, caro,
         saldo, reunio, diretoria, associao, paulista,
         supermercados, (apas), dirigentes, trs, principais
         , cervejarias, (brahma, antarctica, kaiser), \"a,
         deciso, suspender, boicote, no, cabe, apas\", omar
         , assaf, vice-presidente, entidade, referindo-se,
         membros, diretoria, reuniram, cervejarias, apas,
         marcar, assemblia, quinta-feira, associados,
         decidir, continuidade, no, boicote, brahma,
         continua, excluda, lista, compra, supermercados,
         assaf, no, quis, revelar, brahma, convincente,
         explicao, aumentos, preos, corte, descontos,
         motivaram, boicote, fato, indstrias, procurarem,
         apas, explicar, porqu, aumentos, corte, descontos,
          \"j, , satisfaz\", \"no, queramos, assumir, alta,
          brusca, consumidor, preo, cerveja, saltar, cr$,
         20o, cr$, 800, maro, abril\", assaf, prtica,
         supermercadistas, queriam, caracterizar, so, viles
         , aumentos, junto, , opinio, pblica, magin,
         rodrigues, diretor, brahma, entanto, indstrias, no
         , tomaram, iniciativa, marcar, reunio, apas, \"
         apelamos, supermercadistas, no, deixem, consumidor
         , cerveja, brahma\", magin, empresa, no, amargou,
         prejuzos, boicote, redirecionou, venda, bares,
         padarias, supermercados, representam, 30%, vendas,
          cerveja, supermercados, vo, preos, cerveja,
         reajustados, 16, 5%, correo, corresponde, , variao
```

```
          , unidade, real, (urv), 1, 14, abril, descontos,
          variam, at, 17%, supermercadistas, no, fizeram,
          contas, aumento, representar, consumidor, egdio,
          antonio, camillo, gerente, comercial, antarctica,
          esclarece, indstria, est, cumprindo, combinado,
          governo, maio, reajustes, quinzenais, passam,
          semanais, descontos, bateram, casa, 60%, aps,
          carnaval, vo, reduzidos, gradativamente, at, serem
          , zerados, junho, camillo, provavelmente, muitas,
          empresas, vo, fechar, trimestre, vermelho, motivo
          :, vendas, caram, 50%, carnaval]"
   }
}
```

Analisando os resultados percebemos que as duas consultas retornadas foram diferentes das consultas anteriores, com isso podemos perceber que como as consultas retornadas sem o operator, apesar de não possuirem os dois termos, possuiam mais vezes um dos termos, isso fez com que o score obtido fosse mais alto.

4.3. Experimento utilizando minimum_should_match

O minimum_should_match indica um percentual das palavras da consulta que o documento deve conter para ser retornado, por exemplo, se a consulta possui 3 palavras e o minimum_should_match é setado com 66%, somente serão retornados os documentos que possuem pelo menos 2 das 3 palavras da consulta.No exemplo abaixo foi utilizada uma consulta match com o minimum_should_match.

Listing 8. Consulta

```
{
   "query": {
      "match" : {
         "TEXT" : {
            "query": "Boicotes consumidores governo",
            "fuzziness": "AUTO",
            "minimum_should_match": "66%"
         }
      }
   },
   "_source": ["DOCID","DOCNO","DATE","TEXT"],
   "size": 5
}
```

Listing 9. Resultado

```
{
   "_index": "documentos",
   "_type": "documento",
   "_id": "87635",
   "_score": 22.856691,
```

```
   "_source": {
      "DOCID": "FSP951002-033",
      "DATE": "951002",
      "DOCNO": "FSP951002-033",
      "TEXT": "[especial, folha, lisboa, resultado, oficial
         , eleies, portuguesas, s, sair, 15, dias,
         responsabilidade, , pequenas, sees, decidiram,
         boicotar, eleies, boicotes, esto, ligados, questes
         , prticas, gua, esgoto, asfalto, escolas, regies,
         lei, portuguesa, prev, caso, sees, no, consigam,
         promover, votao, tentativas, aps, prazo, dias,
         total, 15, sees, eleitorais, reunindo, cerca, 15,
         eleitores, boicotaram, eleies, boicotes, ganharam,
          fora, eleies, 1991, psd, conquistou, deputado,
         lisboa, vila, santa, maria, decidiu, no, abrir,
         seo, eleitoral, gua, encanada, seo, santa, maria,
         300, eleitores, faltavam, 190, votos, partido,
         socialista, revolucionrio, eleger, deputado, psd,
         conseguiu, manter, boicote, chegou, gua, encanada,
          boicotes, acabaram, legitimados, motivos, boicote
         , vo, mau, cheiro, fbrica, peixe, so, romo,
         coronado, porto, (norte), excesso, buracos,
         estrada, longriva, bragana, (norte), alm, gua,
         benafim, sul, pas, (jr)]"
   }
},
{
   "_index": "documentos",
   "_type": "documento",
   "_id": "16635",
   "_score": 17.096573,
   "_source": {
      "DOCID": "FSP950803-044",
      "DATE": "950803",
      "DOCNO": "FSP950803-044",
      "TEXT": "[vinicius, torres, freire, paris, governo,
         australiano, convocou, estados, compem, pas,
         boicotar, empresas, francesas, medida, , represlia
         , retomada, testes, nucleares, frana, boicotes,
         australianos, abriram, crise, diplomtica, pases,
         frana, retirou, indeterminado, embaixador,
         austrlia, setembro, frana, explodir, bombas, atol,
          mururoa, (polinsia, francesa, oceano, pacfico),
         ministro, defesa, australiano, robert, ray, pediu,
          edf, (eletricidade, frana), proibida, participar,
          privatizao, us$, 9, bilhes, contrato, compagnie,
```

```
          lyonnaise, des, eaux, (de, abastecimento, gua),
          tambm, est, ameaado, anteontem, ray, excluiu,
          dassault, fabricante, francesa, avies, concorrncia
          , us$, 400, milhes, jornal, francs, ``le, monde\",
           dassault, no, chance, alguma, vencer, disputa,
          venda, avies, treinamento, reunio, frum, segurana,
           sia-pacfico, asean, (associao, pases, sudeste,
          asitico), encerrada, emitiu, declarao, pedia,
          testes, nucleares, declarao, no, mencionou, china,
           frana, pases, prosseguem, experincias, atmicas,
          primeiro-ministro, francs, alain, jupp, disse,
          entrevista, coletiva, governo, continua, firme,
          propsito, realizar, testes, seguida, franois,
          barouin, porta-voz, governo, afirmou, ``ofensiva,
          discriminatria\", australiana, aproveita, motivo,
          testes, nucleares, expandir, influncia, oceania,
          sia, ``a, austrlia, , movida, ambies, econmicas,
          geopolticas, regio, frana, forte, presena\", disse
          , barouin, jornal, ``le, parisien\", divulgou,
          pesquisa, 60%, franceses, consideram, chirac,
          desistir, testes, jornal, ``le, quotidien, paris
          \", traz, pgina, convocao, boicote, produtos,
          pases, estariam, discriminando, frana, (austrlia,
          dinamarca, japo, noruega, zelndia), cadeia, hotis,
           sueca, anunciou, no, servir, vinhos, franceses,
          restaurantes, parlamento, governo, suecos, tambm,
          j, haviam, adotado, medida, agncias,
          internacionais]"
    }
}
```

4.4. Experimento utilizando uma consulta booleana

Foi pensado em realizar uma consulta booleana para poder comparar os resultados, utlizando outro tipo de consulta, só que a consulta booleana pode ser reescrita utilizando uma consulta match com o operator and. Abaixo segue uma consulta booleana e a sua consulta match equivalente

Listing 10. Consulta Booleana

```
{
  "bool": {
    "must": [
      { "term": { "TEXT": "Boicote" }},
      { "term": { "TEXT": "consumidores" }}
    ]
  }
}
```

Listing 11. Consulta Match

```
{
  "match": {
    "TEXT": {
       "query" : "Boicote consumidores",
       "operator" : "and"
    }
  }
}
```

Referências

Documentação elasticseach. https://www.elastic.co/guide/en/elasticsearch/reference/5.6/index.html. Online; Acessado Maio 2019.