Projeto - Mineração de texto com stack ELK [22E2_3]

Erik Tavares dos Anjos
Date Updated: 25/06/2022
GIT: [https://github.com/eriktavares/projeto-minera-o-de-texto-com-stack-ELK (https://github.com/eriktavares/projeto-minera-o-de-texto-com-stack-ELK)](https://github.com/eriktavares/projeto-minera-o-de-texto-com-stack-ELK)

Objetivo

A partir de uma instalação de ES e Kibana, realize as seguintes tarefas:
1_Escolha um dataset contendo ao menos 1 campo textual. Descreva os campos do seu dataset e seus respectivos tipos de dado.
2_Elabore um analyzer (pipeline de pré-processamento) de texto para o campo textual. Justifique suas escolhas de tokenizer e token filters.
3_Crie uma configuração de mapping para o índice que receberá o dataset. Realize a importação de dados para esse índice.
4_Crie uma busca com agregação sobre os dados inseridos.
5_Use a query More Like This para realizar a predição de outro campo do seu dataset, como se fosse um kNN.

1. Cenário

O dataset abaixo está disponível no site kaggle pela URL abaixo.

News Category Dataset
Identify the type of news based on headlines and short descriptions
[https://www.kaggle.com/datasets/rmisra/news-category-dataset (https://www.kaggle.com/datasets/rmisra/news-category-dataset)](https://www.kaggle.com/datasets/rmisra/news-category-dataset).

Context
This dataset contains around 200k news headlines from the year 2012 to 2018 obtained from HuffPost. The model trained on this dataset could be used to identify tags for untracked news articles or to identify the type of language used in different news articles.

1.1 - Leitura do Dataset

O dataset tras 6 colunas 'category', 'headline', 'authors', 'link', 'short_description', 'date'.

category: Categória em que a notícia é classificada.
headline: Título da Notícia.
authors: Autores.

link: Link para o site.
short_description: Descrição curta, ou um resumo da notícia.
date: Data da publicação.

In [426]:
```
import pandas as pd
df_data = pd.read_json('../Data/archive5zip/News_Category_Dataset_v2.json', orier
df_data.head(1).T
```

Out[426]:

	0
category	CRIME
headline	There Were 2 Mass Shootings In Texas Last Week...
authors	Melissa Jeltsen
link	https://www.huffingtonpost.com/entry/texas-ama...
short_description	She left her husband. He killed their children...
date	2018-05-26 00:00:00

1.1.1.Categoria

São diversas categórias, como Politica, Bem estar, Enterterimento, Viagens, Stylo e Beleza, e diversas outras. O tipo de dado é um keyword, uma palavra chave.

In [427]:
```
df_data['category'].value_counts()
```

Out[427]:
```
POLITICS          32739
WELLNESS          17827
ENTERTAINMENT     16058
TRAVEL             9887
STYLE & BEAUTY     9649
PARENTING          8677
HEALTHY LIVING     6694
QUEER VOICES       6314
FOOD & DRINK       6226
BUSINESS           5937
COMEDY             5175
SPORTS             4884
BLACK VOICES       4528
HOME & LIVING      4195
PARENTS            3955
THE WORLDPOST      3664
WEDDINGS           3651
WOMEN              3490
IMPACT             3459
DIVORCE            3426
CRIME              3405
MEDIA              2815
WEIRD NEWS         2670
GREEN              2622
WORLDPOST          2579
RELIGION           2556
STYLE              2254
SCIENCE            2178
WORLD NEWS         2177
TASTE              2096
TECH               2082
MONEY              1707
ARTS               1509
FIFTY              1401
GOOD NEWS          1398
ARTS & CULTURE     1339
ENVIRONMENT        1323
COLLEGE            1144
LATINO VOICES      1129
CULTURE & ARTS     1030
EDUCATION          1004
Name: category, dtype: int64
```

1.1.2 Demais Colunas

O **headline** é o título da notícia, abaixo impresso em negrito, e o **short_description** é a descrição cruta, ambos são textos curtos e que foram um resumo do que será o assunto da notícia. Os campos Author, tras o nome do autor da noticia e tem o tipo keywork. A data de publicação está no formato data e o link no formato texto

Abaixo estão alguns exemplos de notícias (5 primeiras do dataset)

In [428]:
```python
BOLD = '\033[1m'
NORMAL = '\033[0m'

for i in range(0,5):
    print("-----------------------------------------------------------------")
    print("Notícia {}:".format(i))
    print(BOLD+df_data['headline'].iloc[i]+'\n'+NORMAL+df_data['short_descriptior
    print(df_data['authors'].iloc[i], df_data['date'].iloc[i])
    print(df_data['link'].iloc[i], '\n')
```

```
----------------------------------------------------------------
Notícia 0:
There Were 2 Mass Shootings In Texas Last Week, But Only 1 On TV
She left her husband. He killed their children. Just another day in America.
Melissa Jeltsen 2018-05-26 00:00:00
https://www.huffingtonpost.com/entry/texas-amanda-painter-mass-shooting_us_5
b081ab4e4b0802d69caad89 (https://www.huffingtonpost.com/entry/texas-amanda-p
ainter-mass-shooting_us_5b081ab4e4b0802d69caad89)

----------------------------------------------------------------
Notícia 1:
Will Smith Joins Diplo And Nicky Jam For The 2018 World Cup's Official Song
Of course it has a song.
Andy McDonald 2018-05-26 00:00:00
https://www.huffingtonpost.com/entry/will-smith-joins-diplo-and-nicky-jam-fo
r-the-official-2018-world-cup-song_us_5b09726fe4b0fdb2aa541201 (https://www.
huffingtonpost.com/entry/will-smith-joins-diplo-and-nicky-jam-for-the-offici
al-2018-world-cup-song_us_5b09726fe4b0fdb2aa541201)

----------------------------------------------------------------
Notícia 2:
Hugh Grant Marries For The First Time At Age 57
The actor and his longtime girlfriend Anna Eberstein tied the knot in a civi
l ceremony.
Ron Dicker 2018-05-26 00:00:00
https://www.huffingtonpost.com/entry/hugh-grant-marries_us_5b09212ce4b0568a8
80b9a8c (https://www.huffingtonpost.com/entry/hugh-grant-marries_us_5b09212c
e4b0568a880b9a8c)

----------------------------------------------------------------
Notícia 3:
Jim Carrey Blasts 'Castrato' Adam Schiff And Democrats In New Artwork
The actor gives Dems an ass-kicking for not fighting hard enough against Don
ald Trump.
Ron Dicker 2018-05-26 00:00:00
https://www.huffingtonpost.com/entry/jim-carrey-adam-schiff-democrats_us_5b0
950e8e4b0fdb2aa53e675 (https://www.huffingtonpost.com/entry/jim-carrey-adam-
schiff-democrats_us_5b0950e8e4b0fdb2aa53e675)

----------------------------------------------------------------
Notícia 4:
Julianna Margulies Uses Donald Trump Poop Bags To Pick Up After Her Dog
The "Dietland" actress said using the bags is a "really cathartic, therapeut
ic moment."
Ron Dicker 2018-05-26 00:00:00
https://www.huffingtonpost.com/entry/julianna-margulies-trump-poop-bag_us_5b
```

```
093ec2e4b0fdb2aa53df70 (https://www.huffingtonpost.com/entry/julianna-margul
ies-trump-poop-bag_us_5b093ec2e4b0fdb2aa53df70)
```

2. Pré Processamento

Esses dados serão migrados para o banco de dados Elasticsearch 8.2.2. para isso será realizado a analise dos campos para posteriormente realizar a criação do mapeamento.

Elasticsearch
o Elasticsearch é um mecanismo de busca e análise de dados distribuído, gratuito e aberto para todos os tipos de dados, incluindo textuais, numéricos, geoespaciais, estruturados e não estruturados. O Elasticsearch é desenvolvido sobre o Apache Lucene e foi lançado pela primeira vez em 2010 pela Elasticsearch N.V. (agora conhecida como Elastic). Conhecido por suas REST APIs simples, natureza distribuída, velocidade e escalabilidade, o Elasticsearch é o componente central do Elastic Stack, um conjunto de ferramentas gratuitas e abertas para ingestão, enriquecimento, armazenamento, análise e visualização de dados. Comumente chamado de ELK Stack (pelas iniciais de Elasticsearch, Logstash e Kibana), o Elastic Stack agora inclui uma rica coleção de agentes lightweight conhecidos como Beats para enviar dados ao Elasticsearch. [https://www.elastic.co/pt/what-is/elasticsearch] (https://www.elastic.co/pt/what-is/elasticsearch%5D)

In [429]:
```
import elasticsearch
import getpass
import urllib3
urllib3.disable_warnings()
senha = getpass.getpass("Digite sua senha: ")

ES_URL = 'https://localhost:9200'
ES_USER = 'elastic'
ES_PASS = senha

client = elasticsearch.Elasticsearch(
    ES_URL,
    basic_auth=(ES_USER, ES_PASS),
    verify_certs=False
)
dict(client.info())
```

```
Digite sua senha: ········

C:\ProgramData\Anaconda3\lib\site-packages\elasticsearch\_sync\client\__init__.
py:395: SecurityWarning: Connecting to 'https://localhost:9200' using TLS with
verify_certs=False is insecure
  _transport = transport_class(
```

Out[429]:
```
{'name': 'DESKTOP-4SAUDI3',
 'cluster_name': 'elasticsearch',
 'cluster_uuid': 'HTrqKXa6SSyS9cJuvdjf4A',
 'version': {'number': '8.2.2',
  'build_flavor': 'default',
  'build_type': 'zip',
  'build_hash': '9876968ef3c745186b94fdabd4483e01499224ef',
  'build_date': '2022-05-25T15:47:06.259735307Z',
  'build_snapshot': False,
  'lucene_version': '9.1.0',
  'minimum_wire_compatibility_version': '7.17.0',
  'minimum_index_compatibility_version': '7.0.0'},
 'tagline': 'You Know, for Search'}
```

2.1 Analizer

Text analysis enables Elasticsearch to perform full-text search, where the search returns all relevant results rather than just exact matches [https://www.elastic.co/guide/en/elasticsearch/reference/8.2/analysis-overview.html](https://www.elastic.co/guide/en/elasticsearch/reference/8.2/analysis-overview.html%5D).

Nesta sessão será realizado a implementação de um analizador para ser aplicado aos campos headline e short_description das noticias.

Para iniciar, será feito o carregamento das stop words em ingles do nltk, essas stop words são palavras que não agregam para resultados de analises de texto.

In [430]:
```
import nltk
stop_words_en=nltk.corpus.stopwords.words('english')
print(' , '.join([x for x in stop_words_en]))
```

```
i , me , my , myself , we , our , ours , ourselves , you , you're , you've , yo
u'll , you'd , your , yours , yourself , yourselves , he , him , his , himself
, she , she's , her , hers , herself , it , it's , its , itself , they , them ,
their , theirs , themselves , what , which , who , whom , this , that , that'll
, these , those , am , is , are , was , were , be , been , being , have , has ,
had , having , do , does , did , doing , a , an , the , and , but , if , or , b
ecause , as , until , while , of , at , by , for , with , about , against , bet
ween , into , through , during , before , after , above , below , to , from , u
p , down , in , out , on , off , over , under , again , further , then , once ,
here , there , when , where , why , how , all , any , both , each , few , more
, most , other , some , such , no , nor , not , only , own , same , so , than ,
too , very , s , t , can , will , just , don , don't , should , should've , now
, d , ll , m , o , re , ve , y , ain , aren , aren't , couldn , couldn't , didn
, didn't , doesn , doesn't , hadn , hadn't , hasn , hasn't , haven , haven't ,
isn , isn't , ma , mightn , mightn't , mustn , mustn't , needn , needn't , shan
, shan't , shouldn , shouldn't , wasn , wasn't , weren , weren't , won , won't
, wouldn , wouldn't
```

Analizer O analiser é o conjunto de char_filter, tokenizer e token filter, de forma simplificada.

Tokenizer Standard tokenizeredit The standard tokenizer provides grammar based tokenization (based on the Unicode Text Segmentation algorithm, as specified in Unicode Standard Annex #29) and works well for most languages [https://www.elastic.co/guide/en/elasticsearch/reference/8.2/analysis-standard-tokenizer.html] (https://www.elastic.co/guide/en/elasticsearch/reference/8.2/analysis-standard-tokenizer.html%5D).

De forma simplificada, os tokens são separados por espaço, não há remoção de número ou de apostrofe. Conforme, exemplo abaixo de um texto do dataset.

In [431]:
```
analyzed = client.indices.analyze(
        tokenizer="standard",
        text="Will Smith Joins Diplo And Nicky Jam For The 2018 World Cup's Offic
)
print([x["token"] for x in analyzed['tokens']])
```

```
['Will', 'Smith', 'Joins', 'Diplo', 'And', 'Nicky', 'Jam', 'For', 'The', '201
8', 'World', "Cup's", 'Official', 'Song']
```

Para remorada dos números será utilizado Char filter.

Char filter Os filtros de caracteres são usados para pré-processar o fluxo de caracteres antes que ele seja passado para o tokenizador. No analisardor desenvolvido, possui um char_filter para remoção de números, como são textos de noticias, são é necessário que os números virem tokens.

Token filter Os token filter ja modificam os tokens após o tokenizer. Para o analizador, será utilizado token filter "lowercase", "asciifolding", "apostrophe" e "stop_custom". O lowercase para passar as letras para minusculas dos tokens. O asciifolding remove acentos e caracteres especiais. O Apostrophe é para remoção de apostrophe dos tokens, como em [cup's] --> [cup]. Por ultimo, o custom stop words para utilização das stop words do nltk.

In [432]:
```
INDICE_NAME = 'category_index'


analysis={
          "analyzer": {
            "analizer_text": {
                  "char_filter": [
                    "replace_numbers"
                    ],
                  "tokenizer": "standard",
                  "filter": [
                      "lowercase",
                      "asciifolding",
                      "apostrophe",
                      "stop_custom",
                  ]
                }
              },
          "char_filter": {
             "replace_numbers": {
                    "type": "pattern_replace",
                    "pattern": "([0-9]+)",
                    "replacement": ""
             },
          },
          "filter": {
            "english_stop": {
              "type": "stop",
              "stopwords": "_english_"
            },
            "stop_custom": {
                "type": "stop",
                "stopwords": stop_words_en,
        },
    }
}

text_category_analizer = {
    "settings": {
        "analysis": analysis
      }
    }

if client.indices.exists(index=INDICE_NAME):
    client.indices.delete(index=INDICE_NAME)
client.indices.create(index=INDICE_NAME, **text_category_analizer)
```

Out[432]:
```
ObjectApiResponse({'acknowledged': True, 'shards_acknowledged': True, 'index':
'category_index'})
```

O resultado do analizador pode ser visto em alguns textos do dataset (5 primeiros), Linha em negrito com headline e embaixo, após uma lista de tokens após o analizer. Na linha normal, logo abaixo, o short description e uma lista com os tokens resultantes.

In [433]:
```
for i in range(0,5):
    print("--------------------------------------------------------------------")
    print("Noticia",i)
    resp_title = client.indices.analyze(
        index="category_index",
        analyzer="analizer_text",
        text=df_data['headline'].iloc[i],
    )
    resp_desc = client.indices.analyze(
        index="category_index",
        analyzer="analizer_text",
        text=df_data['short_description'].iloc[i],
    )
    print(BOLD+df_data['headline'].iloc[i])
    print('[', BOLD+' , '.join([x["token"] for x in resp_title['tokens']]),']\n')

    print(NORMAL+df_data['short_description'].iloc[i])
    print(NORMAL+'[',' , '.join([x["token"] for x in resp_desc['tokens']]),"]")

    #print(df_data['authors'].iloc[i], df_data['date'].iloc[i])
    #print(df_data['link'].iloc[i], '\n')
```

```
---------------------------------------------------------------
Noticia 0
There Were 2 Mass Shootings In Texas Last Week, But Only 1 On TV
[ mass , shootings , texas , last , week , tv ]

She left her husband. He killed their children. Just another day in America.
[ left , husband , killed , children , another , day , america ]
---------------------------------------------------------------
Noticia 1
Will Smith Joins Diplo And Nicky Jam For The 2018 World Cup's Official Song
[ smith , joins , diplo , nicky , jam , world , cup , official , song ]

Of course it has a song.
[ course , song ]
---------------------------------------------------------------
Noticia 2
Hugh Grant Marries For The First Time At Age 57
[ hugh , grant , marries , first , time , age ]

The actor and his longtime girlfriend Anna Eberstein tied the knot in a civi
l ceremony.
[ actor , longtime , girlfriend , anna , eberstein , tied , knot , civil , c
eremony ]
---------------------------------------------------------------
Noticia 3
Jim Carrey Blasts 'Castrato' Adam Schiff And Democrats In New Artwork
[ jim , carrey , blasts , castrato , adam , schiff , democrats , new , artwo
rk ]

The actor gives Dems an ass-kicking for not fighting hard enough against Don
ald Trump.
[ actor , gives , dems , ass , kicking , fighting , hard , enough , donald ,
trump ]
---------------------------------------------------------------
```

```
Noticia 4
Julianna Margulies Uses Donald Trump Poop Bags To Pick Up After Her Dog
[ julianna , margulies , uses , donald , trump , poop , bags , pick , dog ]

The "Dietland" actress said using the bags is a "really cathartic, therapeut
ic moment."
[ dietland , actress , said , using , bags , really , cathartic , therapeuti
c , moment ]
```

3. Mapping e insert

3.1 Mapeando os campos

Os campos do dataset serão mapeados para posteriormente serem inseridos no elastichsearch,

Category Campo que descreve o assunto da noticia, será mapeado com um campo extra do tipo keyword para permitir a busca com nome exato. E para o texto, o analiser desenvolvido acima, para garantir que os tokens para as categorias com as regras descritas cima.

headline Campo com titulo da nóticia, tipo text, e com analiser devolvido acima.

short_description Campo com um resumo da nótica ou um subtitulo, tipo text, e com analiser devolvido acima.

authors Campo texto com o nome do autor, analisador padrão do elastic e keyword para busca com nome exato.

Link Link tipo texto

Data Data tipo data

In [434]:

```
INDEX_NAME = 'news_category_detection'

INDEX_MAPPING = {
        "settings": {
            "number_of_shards": 3,
            "analysis": analysis
    },
    "mappings": {
        "properties": {
            "category": {
                "type": "text",
                "analyzer": "analizer_text",
                "fields": {
                    "raw": {
                        "type": "keyword"
                    }
                }
            },
            "headline": {
                "type": "text",
                "analyzer": "analizer_text",
                "fielddata": True,
                "fielddata_frequency_filter": {
                    "min": 0.01,
                    "min_segment_size": 10,
                    },
            },
            "short_description": {
                "type": "text",
                "analyzer": "analizer_text",
                "fielddata": True,
                "fielddata_frequency_filter": {
                    "min": 0.01,
                    "min_segment_size": 10,
                }
            },
            "authors": {
                "type": "text",
                "fields": {
                    "raw": {
                        "type": "keyword"
                    }
                }
            },
            "link": {
                "type": "text"

            },
            "date": {
                "type": "date"
            },
        }
    }
}
```

```
if client.indices.exists(index=INDEX_NAME):
    client.indices.delete(index=INDEX_NAME)
client.indices.create(index=INDEX_NAME, **INDEX_MAPPING)
```

Out[434]: ObjectApiResponse({'acknowledged': True, 'shards_acknowledged': True, 'index': 'news_category_detection'})

A inserção dos documentos será feit através da tenica bulk

3.2 Inserindo os documentos

In [435]:
```
from elasticsearch.helpers import bulk

def gen_documents(df):
    for line in df.index:
         yield df.iloc[line].to_dict()

def gen_index_actions(documents):
    for doc in documents:
        yield {
            '_op_type': 'index',
            '_index': INDEX_NAME,
            **doc,
        }
```

In [436]:
```

if client.indices.exists(index=INDEX_NAME):
    client.indices.delete(index=INDEX_NAME)
client.indices.create(index=INDEX_NAME, **INDEX_MAPPING)
```

Out[436]: ObjectApiResponse({'acknowledged': True, 'shards_acknowledged': True, 'index': 'news_category_detection'})

In [437]:
```
%%time
documents = gen_documents(df_data)
actions = gen_index_actions(documents)

success, errors = bulk(client, actions)

client.indices.refresh(index=INDEX_NAME)
client.indices.flush(index=INDEX_NAME)
```

```
Wall time: 1min 1s
```

Out[437]: ObjectApiResponse({'_shards': {'total': 6, 'successful': 3, 'failed': 0}})

4. Buscas e Agregações

Com os documentos inseridos é possível então realizar buscar os dados e obter seus resultados. Por exemplo, quem são os autores com maior número de notícias, ou os autores com maior número de notícas.

4.1 Busa pelo campo Autores

In [440]:
```
QUERY = {
    'match_all': {}
}

AGG = {
    'grupos': {
        'terms': {
            'field': 'authors.raw',
            'size': 40,
        }
    }
}
resp = client.search(index=INDEX_NAME, query=QUERY, aggregations=AGG, size=0)
```

In [441]:
```
df_authors_agg = pd.DataFrame(resp['aggregations']['grupos']['buckets'])
df_authors_agg
```

Out[441]:

	key	doc_count
0		36620
1	Lee Moran	2423
2	Ron Dicker	1913
3	Reuters, Reuters	1562
4	Ed Mazza	1322
5	Cole Delbyck	1140
6	Andy McDonald	1068
7	Julia Brucculieri	1059
8	Carly Ledbetter	1054
9	Curtis M. Wong	1020
10	Mary Papenfuss	974
11	Bill Bradley	965
12	Dana Oliver	936
13	David Moye	893
14	Sam Levine	893
15	Michelle Manetti	876
16	Michelle Persad	875
17	Nina Golgowski	868
18	Igor Bobic	866
19	Ellie Krupnick	861
20	Dominique Mosbergen	784
21	Jamie Feldman	772
22	James Michael Nichols	764
23	Caroline Bologna	762
24	Rebecca Adams	753
25	Jenna Amatulli	711
26	Matthew Jacobs	702
27	Ryan Grenoble	698
28	Daniel Marans	669
29	Julie R. Thomson	650
30	Suzy Strutner	650
31	Marina Fang	648
32	Sara Boboltz	637
33	Priscilla Frank	630

	key	doc_count
34	Cavan Sieczkowski	627
35	Hilary Hanson	615
36	Alanna Vagianos	607
37	Paige Lavender	598
38	Rebecca Shapiro	589
39	Antonia Blumberg	584

4.2 Busca pela Categória Politica

Palavras mais comentadas nos títulos na categoria de Política

In [442]:

```
QUERY = {
    'term': {
        'category.raw': 'POLITICS'
    }
}

AGG = {
    'grupos': {
        'terms': {
            'field': 'headline',
            'size': 100,
        },
    }
}
resp = client.search(index=INDEX_NAME, query=QUERY, aggregations=AGG, size=0)
df_politcs_agg_headline = pd.DataFrame(resp['aggregations']['grupos']['buckets'])
df_politcs_agg_headline
```

Out[442]:

	key	doc_count
0	trump	8865
1	donald	2953
2	clinton	1468
3	obama	1422
4	gop	1397
...	...	...
95	orlando	32
96	top	32
97	love	30
98	convention	29
99	food	24

100 rows × 2 columns

Uma wordcloud das palavras da categória de Política

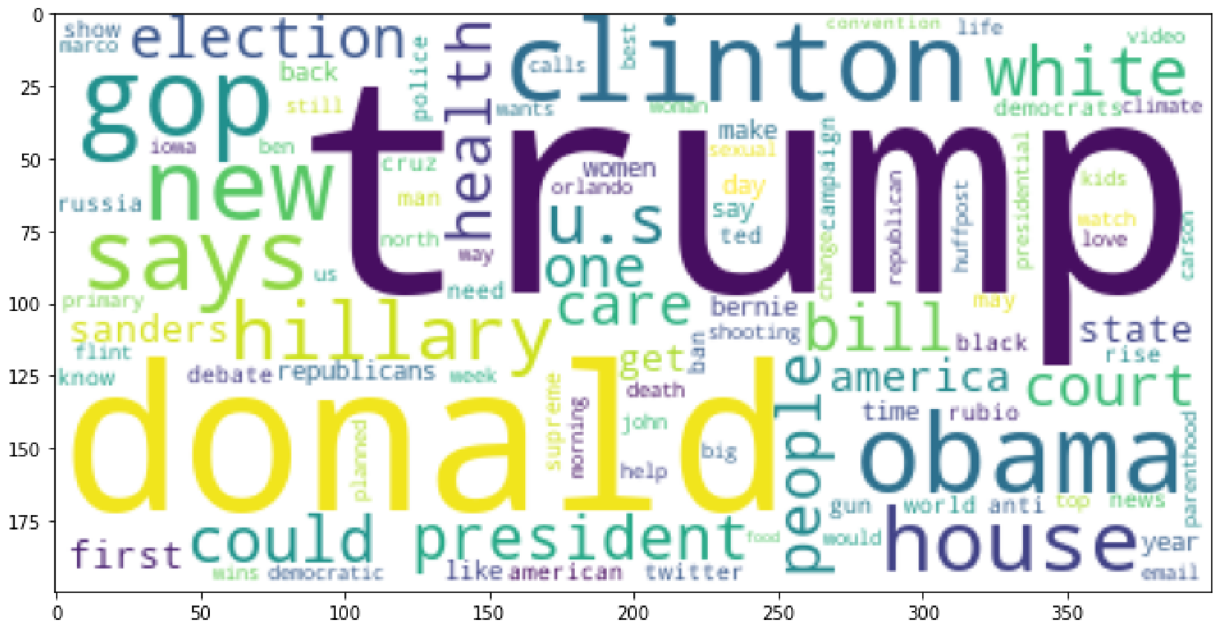
In [443]:

```python
import wordcloud
import matplotlib.pyplot as plt
frequencies = {}
buckets =resp['aggregations']['grupos']['buckets']
for bucket in buckets:
    frequencies[bucket['key']] = bucket['doc_count']

cloud = wordcloud.WordCloud(background_color='white')
cloud.generate_from_frequencies(frequencies)

fig, ax = plt.subplots(figsize=(16, 6))
ax.imshow(cloud)
```

Out[443]: `<matplotlib.image.AxesImage at 0x1d843171490>`



4.3 Busca Geral

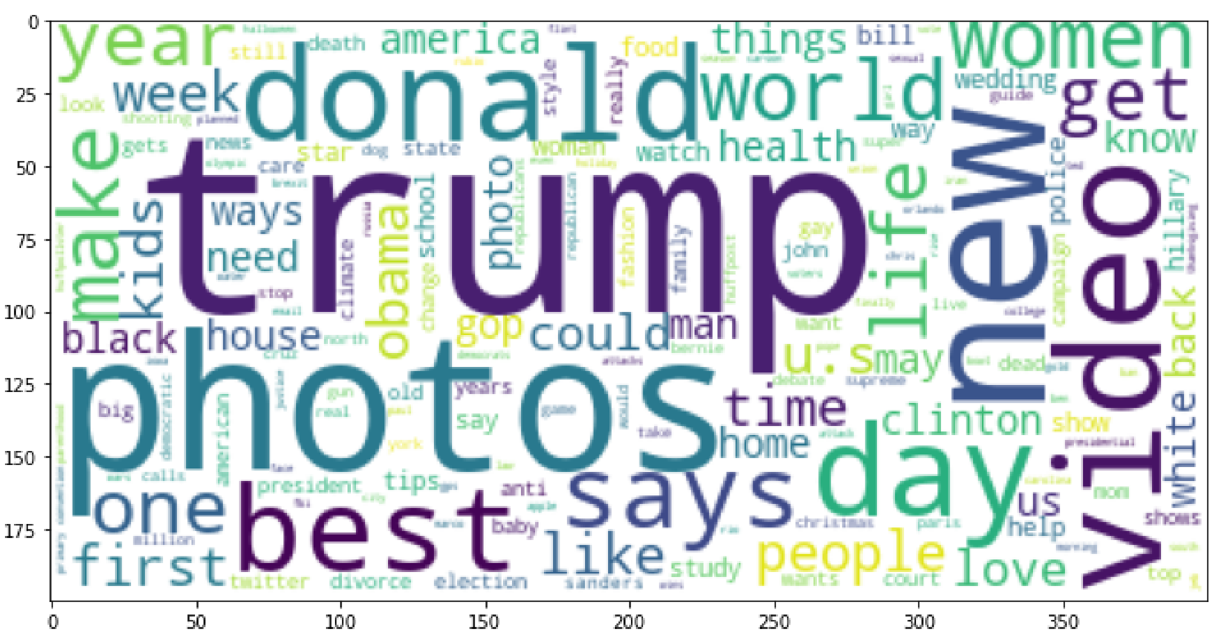
In [444]:
```
QUERY = {
     'match_all': {}
}

AGG = {
    'grupos': {
        'terms': {
            'field': 'headline',
            'size': 2000000
        },
    }
}
resp = client.search(index=INDEX_NAME, query=QUERY, aggregations=AGG, size=0)
frequencies = {}
buckets =resp['aggregations']['grupos']['buckets']
for bucket in buckets:
    frequencies[bucket['key']] = bucket['doc_count']

cloud = wordcloud.WordCloud(background_color='white')
cloud.generate_from_frequencies(frequencies)

fig, ax = plt.subplots(figsize=(16, 6))
ax.imshow(cloud)
```

Out[444]: <matplotlib.image.AxesImage at 0x1d848ea2160>



4. More Like This

The More Like This Query finds documents that are "like" a given set of documents. In order to do so, MLT selects a set of representative terms of these input documents, forms a query using these terms, executes the query and returns the results. The user controls the input documents, how the terms should be selected and how the query is formed [https://www.elastic.co/guide/en/elasticsearch/reference/8.2/query-dsl-mlt-query.html](https://www.elastic.co/guide/en/elasticsearch/reference/8.2/query-dsl-mlt-query.html%5D).

Pode ser utilizada por exemplo para encontrar informações aproximadas, por exemplo, a busca por "president"

In [445]:

```
QUERY = {
    'more_like_this': {
        'fields': ['headline', 'short_description'],
        'like':"president",
        'min_term_freq': 1,
        'max_query_terms': 12,
    }
}
resp = client.search(index=INDEX_NAME, query=QUERY, size=10)
df_resp = pd.DataFrame(x['_source'] for x in resp['hits']['hits'])
df_resp
```

Out[445]:

	category	headline	authors	link	short_descr
0	POLITICS	Not Even Donald Trump Can Believe He's President	Igor Bobic	https://www.huffingtonpost.com/entry/donald-tr...	"I'm presid hey, I'm presi
1	POLITICS	Wednesday's Morning Email: Why Obama May Be Kn...	Lauren Weber	https://www.huffingtonpost.com/entry/wednesday...	The preside commuted sentence
2	POLITICS	Emmanuel Macron Dropped Onto A Nuclear Sub And...	Lee Moran	https://www.huffingtonpost.com/entry/emmanuel-...	"Now th Presi
3	POLITICS	How Obama's 'Brutal' First Job Inspired A New ...	Chris D'Angelo	https://www.huffingtonpost.com/entry/obama-fir...	Before C was presic the United S
4	COMEDY	Stephen Colbert Just Wants Donald Trump's Lawy...	Ron Dicker	https://www.huffingtonpost.com/entry/stephen-c...	He cal Sekulo President Tru
5	POLITICS	Nancy Pelosi Calls For Sean Spicer's Ouster Am...	Igor Bobic	https://www.huffingtonpost.com/entry/nancy-pel...	"Eithe speaking t president,
6	POLITICS	Not Even Mike Pence Can Defend Trump's Wiretap...	Sam Levine	https://www.huffingtonpost.com/entry/mike-penc...	"I thi president's speaks for
7	POLITICS	Obama Urges Russia To Stop Bombing 'Moderate' ...		https://www.huffingtonpost.com/entry/obama-rus...	The pre spok Russian Pre
8	COMEDY	John Oliver Announces His Endorsements For Thi...	Lee Moran	https://www.huffingtonpost.com/entry/john-oliv...	"But r presic co

	category	headline	authors	link	short_descr
9	POLITICS	Tuesday's Morning Email: Everything You Need T...	Lauren Weber	https://www.huffingtonpost.com/entry/tuesdays-...	And what it r for Pre

In [446]:
```
df_resp.iloc[0].T
```

Out[446]:
```
category                                                    POLITICS
headline                Not Even Donald Trump Can Believe He's President
authors                                                   Igor Bobic
link                 https://www.huffingtonpost.com/entry/donald-tr... (http
s://www.huffingtonpost.com/entry/donald-tr...)
short_description           "I’m president — hey, I’m president!"
date                                             2017-05-04T00:00:00
Name: 0, dtype: object
```

In [447]:
```
df_resp ['headline'].iloc[0]
```

Out[447]:
```
"Not Even Donald Trump Can Believe He's President"
```

Classificador KNN com More like this

Os metodos abaixo criam o classificador da seguinte forma, inicialmente seleciona uma quantidade de documentos. Para utilizar o more like this para obter uma quantidade de documentos visinhos e classificar com base na categória do documento selecionado, com base na categória com maior score dos vizinhos.

Implementando Classificador

Seleciona um documento, por exemplo:

In [517]:
```
test_docs_resp = client.search(index=INDEX_NAME, size=1)
test_docs_resp['hits']
```

Out[517]:
```
{'total': {'value': 10000, 'relation': 'gte'},
 'max_score': 1.0,
 'hits': [{'_index': 'news_category_detection',
   '_id': 'rdJ-m4EBE8fvlEs2Vm-w',
   '_score': 1.0,
   '_source': {'category': 'THE WORLDPOST',
    'headline': 'British Mosques Open Doors For Tea And Interfaith Engagement',
    'authors': 'Carol Kuruvilla',
    'link': 'https://www.huffingtonpost.com/entry/visit-my-mosque-day_us_5898aa
ace4b0406131381022',
    'short_description': 'In a climate of rising anti-Muslim rhetoric, the U.K.
held its biggest "Visit My Mosque" day yet.',
    'date': '2017-02-06T00:00:00'}}]}
```

Cria um dataframe para os documentos de teste

In [518]:
```
test_docs = pd.DataFrame(
        {
            '_id': x['_id'],
            **x['_source']
        } for x in test_docs_resp['hits']['hits']
    )
test_docs['category'].value_counts()
```

Out[518]:
```
THE WORLDPOST    1
Name: category, dtype: int64
```

Aplica a query Mode Like This para selecionar os 10 mais próximos deste documento.

In [521]:
```
QUERY = {
    'more_like_this': {
        'fields': ["headline"],
        'like': [
            {
                '_index': INDEX_NAME,
                '_id': 'rdJ-m4EBE8fvlEs2Vm-w',
            }
        ],
        'min_term_freq': 1,
        'max_query_terms': 12,
        'minimum_should_match': -100,
    }
}
resp = client.search(index=INDEX_NAME, query=QUERY, size=10)
resp_df = pd.DataFrame({'_id': x['_id'], '_score': x['_score'], **x['_source']} f
resp_df.groupby('category').sum()
```

Out[521]:

	_score
category	
BUSINESS	12.120592
COLLEGE	12.814263
COMEDY	28.096938
POLITICS	12.132198
RELIGION	30.615908
SPORTS	25.283963
WELLNESS	14.264799

Ordena a lista para que a categória com maior Score fique primeiro

In [522]:
```
resp_df.groupby('category').sum().sort_values('_score', ascending=False)
```

Out[522]:

category	_score
RELIGION	30.615908
COMEDY	28.096938
SPORTS	25.283963
WELLNESS	14.264799
COLLEGE	12.814263
POLITICS	12.132198
BUSINESS	12.120592

Retorna a categória com maior Score

In [524]:
```
resp_df.groupby('category').sum().sort_values('_score', ascending=False).iloc[0]
```

Out[524]:
```
_score    30.615908
Name: RELIGION, dtype: float64
```

Avaliando com mais documentos

Metodo para realização dessa classificação para um conjunto de documentos. Para este dataset foi observado que diversas vezes ele não encontra nenhum documento, então 'minimum_should_match': -100 foi ajustado para ampliar os match com os documentos e evitar erro quando o metodo não retorna nada. E também foi criada uma busca alternativa para caso não retornar nada. Porme mesmo com diversos tipos de buscar como 'fields': ["headline^2","short_description"] e outras tentativas, ainda sim ocorre de não ter nenhum match na consulta like, para este dataset.

In [529]:
```python
def classify_document(doc_id, size=10):
    return classify_with_score(
        [
            {
                '_index': INDEX_NAME,
                '_id': doc_id
            }
        ]
    )

def nova_busca(like, size=10):
    query = {
        'more_like_this': {
            'fields': ["short_description"],
            'like':like,
            'min_term_freq': 1,
            'max_query_terms': 12,
            'minimum_should_match': -100,
        }
    }
    resp = client.search(index=INDEX_NAME, query=query, size=size)
    return formata_retorno(resp)


def formata_retorno(resp):
    resp_df = pd.DataFrame(
                {
                    '_id': x['_id'],
                    '_score': x['_score'],
                    **x['_source']
                } for x in resp['hits']['hits']
            )
    return resp_df.groupby('category').sum().sort_values('_score', ascending=Fals


def classify_with_score(like, size=10):

    query = {
        'more_like_this': {
            'fields': ["headline"],
            'like':like,
            'min_term_freq': 1,
            'max_query_terms': 12,
            'minimum_should_match': -100,
        }
    }

    resp = client.search(index=INDEX_NAME, query=query, size=size)
    if(resp['hits']['total']['value']<0):
        return formata_retorno(resp)
    else:
        return nova_busca(like)


def resp_to_dataFrame(resp):
    test_docs = pd.DataFrame(
```

```
        {
            '_id': x['_id'],
            **x['_source']
        } for x in test_docs_resp['hits']['hits']
    )
    return test_docs
```

In [532]:
```
test_docs_resp = client.search(index=INDEX_NAME, size=50)
test_docs=resp_to_dataFrame(resp)
```

Aplica o metodo para classify_document que chama o metodo classify_with_score que executa o uma consulta more like this passando o id do documento no like. No metodo vai executar a consulta passando um id, receber os 10 vizinhos mais proximos, criar um dataframe com o resultado e agrupar pelas categorias somando os scores.

In [533]:
```
test_docs['predicted'] = test_docs['_id'].apply(classify_document)
```

Um pequeno ajuste precisou ser feito no metodo, primeiro, ordenar a o dataframe pelo score, para que o registro com maior score fique primeiro. E posteriomente retornar a categoria com maior score.

Com as classification_report calcular as metricas de resultados. Os resultados não foram bons, o dataframe possui muitas catgorias e as informações textuais são curtas para se definir.

Os resultados obtivos com a classificação foram muito baixos, mas porque o dataset possui muito categórias, que são muito semelhantes até como por exemplo: WORLD NEWS, THE WORLDPOST e WORLDPOST. Outro caso são buscas onde a consulta like não retorna nenhum documento.

In [534]:
```python
from sklearn.metrics import classification_report
print(
    classification_report(
        test_docs['category'],
        test_docs['predicted']
    )
)
```

```
                precision    recall  f1-score   support

  BLACK VOICES       1.00      0.50      0.67         2
      BUSINESS       0.00      0.00      0.00         1
        COMEDY       0.00      0.00      0.00         2
         CRIME       1.00      1.00      1.00         1
 ENTERTAINMENT       0.33      0.43      0.38         7
     GOOD NEWS       0.00      0.00      0.00         0
         GREEN       0.00      0.00      0.00         1
HEALTHY LIVING       0.00      0.00      0.00         1
 LATINO VOICES       0.00      0.00      0.00         1
         MEDIA       0.00      0.00      0.00         1
     PARENTING       0.00      0.00      0.00         0
       PARENTS       0.00      0.00      0.00         1
      POLITICS       0.46      0.73      0.56        15
  QUEER VOICES       0.00      0.00      0.00         4
      RELIGION       0.00      0.00      0.00         3
       SCIENCE       0.00      0.00      0.00         0
        SPORTS       0.25      0.33      0.29         3
         STYLE       0.00      0.00      0.00         1
STYLE & BEAUTY       0.00      0.00      0.00         0
 THE WORLDPOST       1.00      0.33      0.50         3
        TRAVEL       0.00      0.00      0.00         1
      WELLNESS       0.00      0.00      0.00         0
         WOMEN       1.00      0.50      0.67         2
    WORLD NEWS       0.00      0.00      0.00         0

      accuracy                           0.38        50
     macro avg       0.21      0.16      0.17        50
  weighted avg       0.36      0.38      0.34        50


C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.p
y:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and be
ing set to 0.0 in labels with no predicted samples. Use `zero_division` para
meter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.p
y:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and being
set to 0.0 in labels with no true samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.p
y:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and be
ing set to 0.0 in labels with no predicted samples. Use `zero_division` para
meter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.p
```

```
y:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and being
set to 0.0 in labels with no true samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.p
y:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and be
ing set to 0.0 in labels with no predicted samples. Use `zero_division` para
meter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_classification.p
y:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and being
set to 0.0 in labels with no true samples. Use `zero_division` parameter to
control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

4.2 Outro Dataset

Na avaliação do classificador feita acima, foi observado que os resultados não de f1 e acurácia não foram satisfatorios. Para ter certeza se o problema é o classificador ou o dataset, podemos realizar um teste com outro dataset com numero bem menor de categórias para as noticias.

In [535]:
```
import pandas as pd
df2_data = pd.read_csv('../Data/archive6/bbc-news-data.csv', sep='\t')
df2_data.head(1).T
```

Out[535]:

	0
category	business
filename	001.txt
title	Ad sales boost Time Warner profit
content	Quarterly profits at US media giant TimeWarne...

Esse dataset possui 5 categórias conforme mostrado abaixo:

In [536]:
```
df2_data['category'].value_counts()
```

Out[536]:
```
sport            511
business         510
politics         417
tech             401
entertainment    386
Name: category, dtype: int64
```

In [537]:
```
BOLD = '\033[1m'
NORMAL = '\033[0m'

for i in range(0,5):
    print("------------------------------------------------------------------")
    print("Notícia {}:".format(i))
    print(BOLD+df2_data['title'].iloc[i]+'\n'+NORMAL+df2_data['content'].iloc[i])
    print(df2_data['category'].iloc[i])
```

```
------------------------------------------------------------------
Notícia 0:
Ad sales boost Time Warner profit
 Quarterly profits at US media giant TimeWarner jumped 76% to $1.13bn (£600m) f
or the three months to December, from $639m year-earlier.  The firm, which is n
ow one of the biggest investors in Google, benefited from sales of high-speed i
nternet connections and higher advert sales. TimeWarner said fourth quarter sal
es rose 2% to $11.1bn from $10.9bn. Its profits were buoyed by one-off gains wh
ich offset a profit dip at Warner Bros, and less users for AOL.  Time Warner sa
id on Friday that it now owns 8% of search-engine Google. But its own internet
 business, AOL, had has mixed fortunes. It lost 464,000 subscribers in the four
th quarter profits were lower than in the preceding three quarters. However, th
e company said AOL's underlying profit before exceptional items rose 8% on the
 back of stronger internet advertising revenues. It hopes to increase subscribe
rs by offering the online service free to TimeWarner internet customers and wil
l try to sign up AOL's existing customers for high-speed broadband. TimeWarner
 also has to restate 2000 and 2003 results following a probe by the US Securiti
es Exchange Commission (SEC), which is close to concluding.  Time Warner's four
th quarter profits were slightly better than analysts' expectations. But its fi
lm division saw profits slump 27% to $284m, helped by box-office flops Alexande
r and Catwoman, a sharp contrast to year-earlier, when the third and final film
in the Lord of the Rings trilogy boosted results. For the full-year, TimeWarner
posted a profit of $3.36bn, up 27% from its 2003 performance, while revenues gr
ew 6.4% to $42.09bn. "Our financial performance was strong, meeting or exceedin
g all of our full-year objectives and greatly enhancing our flexibility," chair
man and chief executive Richard Parsons said. For 2005, TimeWarner is projectin
g operating earnings growth of around 5%, and also expects higher revenue and w
ider profit margins.  TimeWarner is to restate its accounts as part of efforts
 to resolve an inquiry into AOL by US market regulators. It has already offered
to pay $300m to settle charges, in a deal that is under review by the SEC. The
 company said it was unable to estimate the amount it needed to set aside for l
egal reserves, which it previously set at $500m. It intends to adjust the way i
t accounts for a deal with German music publisher Bertelsmann's purchase of a s
take in AOL Europe, which it had reported as advertising revenue. It will now b
ook the sale of its stake in AOL Europe as a loss on the value of that stake.
business
------------------------------------------------------------------
Notícia 1:
Dollar gains on Greenspan speech
 The dollar has hit its highest level against the euro in almost three months a
fter the Federal Reserve head said the US trade deficit is set to stabilise.  A
nd Alan Greenspan highlighted the US government's willingness to curb spending
 and rising household savings as factors which may help to reduce it. In late t
rading in New York, the dollar reached $1.2871 against the euro, from $1.2974 o
n Thursday. Market concerns about the deficit has hit the greenback in recent m
onths. On Friday, Federal Reserve chairman Mr Greenspan's speech in London ahea
d of the meeting of G7 finance ministers sent the dollar higher after it had ea
```

rlier tumbled on the back of worse-than-expected US jobs data. "I think the cha irman's taking a much more sanguine view on the current account deficit than h e's taken for some time," said Robert Sinche, head of currency strategy at Bank of America in New York. "He's taking a longer-term view, laying out a set of co nditions under which the current account deficit can improve this year and nex t." Worries about the deficit concerns about China do, however, remain. Chin a's currency remains pegged to the dollar and the US currency's sharp falls in recent months have therefore made Chinese export prices highly competitive. Bu t calls for a shift in Beijing's policy have fallen on deaf ears, despite recen t comments in a major Chinese newspaper that the "time is ripe" for a loosening of the peg. The G7 meeting is thought unlikely to produce any meaningful moveme nt in Chinese policy. In the meantime, the US Federal Reserve's decision on 2 F ebruary to boost interest rates by a quarter of a point - the sixth such move i n as many months - has opened up a differential with European rates. The half-p oint window, some believe, could be enough to keep US assets looking more attra ctive, and could help prop up the dollar. The recent falls have partly been the result of big budget deficits, as well as the US's yawning current account gap, both of which need to be funded by the buying of US bonds and assets by foreign firms and governments. The White House will announce its budget on Monday, and many commentators believe the deficit will remain at close to half a trillion dollars.
business

Notícia 2:
Yukos unit buyer faces loan claim
The owners of embattled Russian oil giant Yukos are to ask the buyer of its fo rmer production unit to pay back a $900m (£479m) loan. State-owned Rosneft bou ght the Yugansk unit for $9.3bn in a sale forced by Russia to part settle a $2 7.5bn tax claim against Yukos. Yukos' owner Menatep Group says it will ask Rosn eft to repay a loan that Yugansk had secured on its assets. Rosneft already fac es a similar $540m repayment demand from foreign banks. Legal experts said Rosn eft's purchase of Yugansk would include such obligations. "The pledged assets a re with Rosneft, so it will have to pay real money to the creditors to avoid se izure of Yugansk assets," said Moscow-based US lawyer Jamie Firestone, who is n ot connected to the case. Menatep Group's managing director Tim Osborne told th e Reuters news agency: "If they default, we will fight them where the rule of l aw exists under the international arbitration clauses of the credit." Rosneft officials were unavailable for comment. But the company has said it intends to take action against Menatep to recover some of the tax claims and debts owed by Yugansk. Yukos had filed for bankruptcy protection in a US court in an attempt to prevent the forced sale of its main production arm. The sale went ahead in December and Yugansk was sold to a little-known shell company which in turn wa s bought by Rosneft. Yukos claims its downfall was punishment for the political ambitions of its founder Mikhail Khodorkovsky and has vowed to sue any particip ant in the sale.
business

Notícia 3:
High fuel prices hit BA's profits
British Airways has blamed high fuel prices for a 40% drop in profits. Report ing its results for the three months to 31 December 2004, the airline made a pr e-tax profit of £75m ($141m) compared with £125m a year earlier. Rod Eddington, BA's chief executive, said the results were "respectable" in a third quarter wh en fuel costs rose by £106m or 47.3%. BA's profits were still better than marke t expectation of £59m, and it expects a rise in full-year revenues. To help of fset the increased price of aviation fuel, BA last year introduced a fuel surch arge for passengers. In October, it increased this from £6 to £10 one-way for

all long-haul flights, while the short-haul surcharge was raised from £2.50 to £4 a leg. Yet aviation analyst Mike Powell of Dresdner Kleinwort Wasserstein sa ys BA's estimated annual surcharge revenues - £160m - will still be way short o f its additional fuel costs - a predicted extra £250m. Turnover for the quarter was up 4.3% to £1.97bn, further benefiting from a rise in cargo revenue. Lookin g ahead to its full year results to March 2005, BA warned that yields - average revenues per passenger - were expected to decline as it continues to lower pric es in the face of competition from low-cost carriers. However, it said sales wo uld be better than previously forecast. "For the year to March 2005, the total revenue outlook is slightly better than previous guidance with a 3% to 3.5% im provement anticipated," BA chairman Martin Broughton said. BA had previously fo recast a 2% to 3% rise in full-year revenue. It also reported on Friday that p assenger numbers rose 8.1% in January. Aviation analyst Nick Van den Brul of BN P Paribas described BA's latest quarterly results as "pretty modest". "It is qu ite good on the revenue side and it shows the impact of fuel surcharges and a p ositive cargo development, however, operating margins down and cost impact of f uel are very strong," he said. Since the 11 September 2001 attacks in the Unite d States, BA has cut 13,000 jobs as part of a major cost-cutting drive. "Our fo cus remains on reducing controllable costs and debt whilst continuing to invest in our products," Mr Eddington said. "For example, we have taken delivery of si x Airbus A321 aircraft and next month we will start further improvements to our Club World flat beds." BA's shares closed up four pence at 274.5 pence.
business

Notícia 4:
Pernod takeover talk lifts Domecq
Shares in UK drinks and food firm Allied Domecq have risen on speculation that it could be the target of a takeover by France's Pernod Ricard. Reports in the Wall Street Journal and the Financial Times suggested that the French spirits f irm is considering a bid, but has yet to contact its target. Allied Domecq shar es in London rose 4% by 1200 GMT, while Pernod shares in Paris slipped 1.2%. Pe rnod said it was seeking acquisitions but refused to comment on specifics. Per nod's last major purchase was a third of US giant Seagram in 2000, the move whi ch propelled it into the global top three of drinks firms. The other two-thirds of Seagram was bought by market leader Diageo. In terms of market value, Pernod - at 7.5bn euros ($9.7bn) - is about 9% smaller than Allied Domecq, which has a capitalisation of £5.7bn ($10.7bn; 8.2bn euros). Last year Pernod tried to buy Glenmorangie, one of Scotland's premier whisky firms, but lost out to luxury g oods firm LVMH. Pernod is home to brands including Chivas Regal Scotch whisky, Havana Club rum and Jacob's Creek wine. Allied Domecq's big names include Mali bu rum, Courvoisier brandy, Stolichnaya vodka and Ballantine's whisky - as well as snack food chains such as Dunkin' Donuts and Baskin-Robbins ice cream. The W SJ said that the two were ripe for consolidation, having each dealt with proble matic parts of their portfolio. Pernod has reduced the debt it took on to fund the Seagram purchase to just 1.8bn euros, while Allied has improved the perfor mance of its fast-food chains.
business

Analiser

Será aproveitado o mesmo analiser feito acima para o 1° dataset.

In [540]:

```python
for i in range(0,5):
    print("-----------------------------------------------------------------")
    print("Noticia",i)
    resp_title = client.indices.analyze(
        index=INDICE_NAME,
        analyzer="analizer_text",
        text=df2_data['title'].iloc[i],
    )
    resp_desc = client.indices.analyze(
        index=INDICE_NAME,
        analyzer="analizer_text",
        text=df2_data['content'].iloc[i],
    )
    print(BOLD+df2_data['title'].iloc[i])
    print('[', BOLD+' , '.join([x["token"] for x in resp_title['tokens']]),']\n')

    print(NORMAL+df2_data['content'].iloc[i], '\n')
    print(NORMAL+'[',' , '.join([x["token"] for x in resp_desc['tokens']]),"]")

    #print(df_data['authors'].iloc[i], df_data['date'].iloc[i])
    #print(df_data['link'].iloc[i], '\n')
```

```
-----------------------------------------------------------------
Noticia 0
Ad sales boost Time Warner profit
[ ad , sales , boost , time , warner , profit ]

 Quarterly profits at US media giant TimeWarner jumped 76% to $1.13bn (£600
m) for the three months to December, from $639m year-earlier.  The firm, whi
ch is now one of the biggest investors in Google, benefited from sales of hi
gh-speed internet connections and higher advert sales. TimeWarner said fourt
h quarter sales rose 2% to $11.1bn from $10.9bn. Its profits were buoyed by
 one-off gains which offset a profit dip at Warner Bros, and less users for
 AOL.  Time Warner said on Friday that it now owns 8% of search-engine Googl
e. But its own internet business, AOL, had has mixed fortunes. It lost 464,0
00 subscribers in the fourth quarter profits were lower than in the precedin
g three quarters. However, the company said AOL's underlying profit before e
xceptional items rose 8% on the back of stronger internet advertising revenu
es. It hopes to increase subscribers by offering the online service free to
 TimeWarner internet customers and will try to sign up AOL's existing custom
ers for high-speed broadband. TimeWarner also has to restate 2000 and 2003 r
```

Mapeamento

In [553]:
```
INDEX_NAME = 'news_category_detection_data_2'

INDEX_MAPPING = {
        "settings": {
            "number_of_shards": 3,
            "analysis": analysis
    },
    "mappings": {
        "properties": {
            "category": {
                "type": "text",
                "analyzer": "analizer_text",
                "fields": {
                    "raw": {
                        "type": "keyword"
                    }
                }
            },
            "title": {
                "type": "text",
                "analyzer": "analizer_text",
                "fielddata": True,
                "fielddata_frequency_filter": {
                    "min": 0.01,
                    "min_segment_size": 10,
                    },
            },
            "content": {
                "type": "text",
                "analyzer": "analizer_text",
                "fielddata": True,
                "fielddata_frequency_filter": {
                    "min": 0.01,
                    "min_segment_size": 10,
                }
            },
            "filename": {
                "type": "text",
                "fields": {
                    "raw": {
                        "type": "keyword"
                    }
                }
            },
        }
    }
}

if client.indices.exists(index=INDEX_NAME):
    client.indices.delete(index=INDEX_NAME)
client.indices.create(index=INDEX_NAME, **INDEX_MAPPING)
```

Out[553]:
```
ObjectApiResponse({'acknowledged': True, 'shards_acknowledged': True, 'index':
'news_category_detection_data_2'})
```

Inserção dos dados no elastich

In [554]:
```
%%time
documents = gen_documents(df2_data)
actions = gen_index_actions(documents)

success, errors = bulk(client, actions)

client.indices.refresh(index=INDEX_NAME)
client.indices.flush(index=INDEX_NAME)
```

```
Wall time: 1.6 s
```

Out[554]:
```
ObjectApiResponse({'_shards': {'total': 6, 'successful': 3, 'failed': 0}})
```

Nuvem de palavras

Palavras como UK (united Kingdon) e US (United States), new (pode ser inserida na lista de stop words, por ventura), blair e film estão entre as que mais aparecem no titulo.

In [556]:

```python
INDEX_NAME = 'news_category_detection_data_2'
QUERY = {
     'match_all': {}
}

AGG = {
    'grupos': {
        'terms': {
            'field': 'title',
            'size': 22000
        },
    }
}
resp = client.search(index=INDEX_NAME, query=QUERY, aggregations=AGG, size=0)
frequencies = {}
buckets =resp['aggregations']['grupos']['buckets']
for bucket in buckets:
    frequencies[bucket['key']] = bucket['doc_count']

cloud = wordcloud.WordCloud(background_color='white')
cloud.generate_from_frequencies(frequencies)

fig, ax = plt.subplots(figsize=(16, 6))
ax.imshow(cloud)
resp['hits']
```

Out[556]:

```
{'total': {'value': 2225, 'relation': 'eq'}, 'max_score': None, 'hits': []}
```

Nuvem de palavras do texto

In [557]:

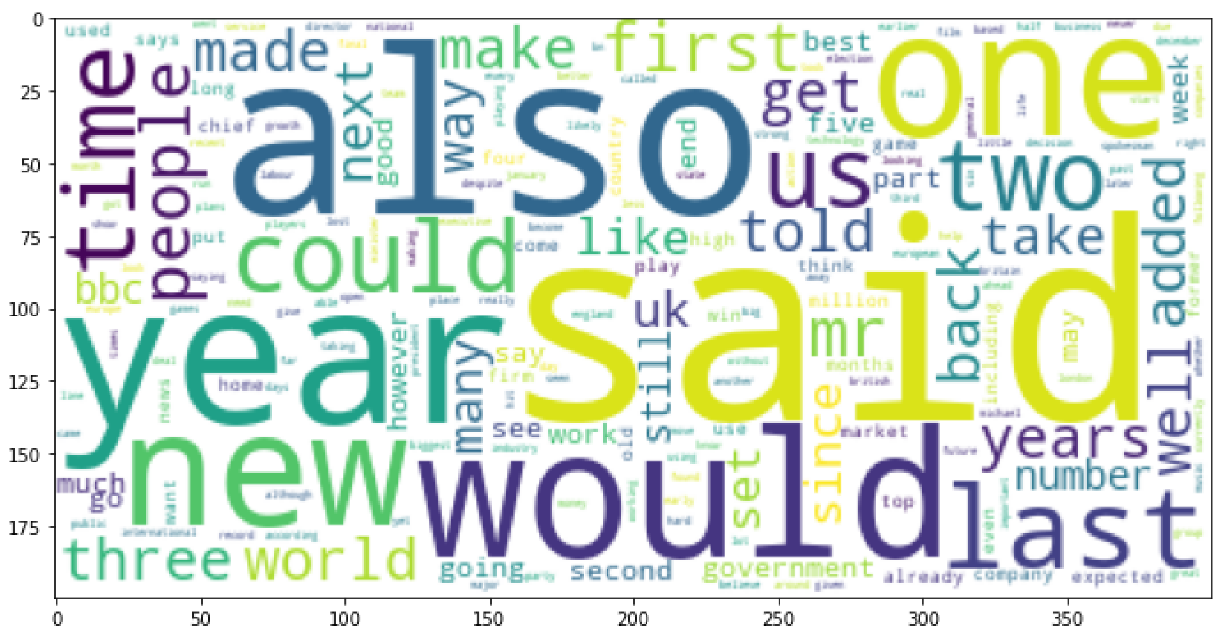
```
QUERY = {
     'match_all': {}
}

AGG = {
    'grupos': {
        'terms': {
            'field': 'content',
            'size': 2000000
        },
    }
}
resp = client.search(index=INDEX_NAME, query=QUERY, aggregations=AGG, size=0)
frequencies = {}
buckets =resp['aggregations']['grupos']['buckets']
for bucket in buckets:
    frequencies[bucket['key']] = bucket['doc_count']

cloud = wordcloud.WordCloud(background_color='white')
cloud.generate_from_frequencies(frequencies)

fig, ax = plt.subplots(figsize=(16, 6))
ax.imshow(cloud)
```

Out[557]: <matplotlib.image.AxesImage at 0x1d843c7eaf0>



In [559]:
```
Classificador para o Dataset novo.
```

```
  File "C:\Users\Erik\AppData\Local\Temp/ipykernel_20388/4031211841.py", line 1
    Classificador para o Dataset novo.
                  ^
SyntaxError: invalid syntax
```

In [560]:
```python
def classify_document2(doc_id, size=10):
    return classify_with_score2(
        [
            {
                '_index': INDEX_NAME,
                '_id': doc_id
            }
        ]
    )

def classify_with_score2(like, size=10):

    query = {
        'more_like_this': {
            'fields': ["content"],
            'like':like,
            'min_term_freq': 1,
            'max_query_terms': 12,
            'minimum_should_match': -100,
        }
    }
    resp = client.search(index=INDEX_NAME, query=query, size=size)
    resp_df = pd.DataFrame(
            {
                '_id': x['_id'],
                '_score': x['_score'],
                **x['_source']
            } for x in resp['hits']['hits']
        )


    return resp_df.groupby('category').sum().sort_values('_score', ascending=Fals
```

Avaliação do Classificador

In [561]:
```
test_docs_resp = client.search(index=INDEX_NAME, size=1000, _source=['category']]
test_docs = pd.DataFrame({'_id': x['_id'],**x['_source']} for x in test_docs_resp
test_docs['predicted'] = test_docs['_id'].apply(classify_document2)
print(classification_report(test_docs['category'], test_docs['predicted']))
```

```
               precision    recall  f1-score   support

     business       0.98      0.93      0.96       326
entertainment       0.99      0.94      0.96       211
     politics       0.89      0.97      0.93       158
        sport       0.97      0.99      0.98       177
         tech       0.91      0.98      0.94       128

     accuracy                           0.96      1000
    macro avg       0.95      0.96      0.95      1000
 weighted avg       0.96      0.96      0.96      1000
```

Acurácia de 94%, resultado muito bom para a definição nesses 5 temas através dos 10 vizinhos mais póximos. Esse segundo dataset acaba sendo melhor para se classificar dessa forma, por que possui o campo content que tem textos mais longos, e menos categórias do que o primeiro.

In [562]:
```
test_docs_resp = client.search(index=INDEX_NAME, size=2225, _source=['category']]
test_docs = pd.DataFrame({'_id': x['_id'],**x['_source']} for x in test_docs_resp
test_docs['predicted'] = test_docs['_id'].apply(classify_document2)
print(classification_report(test_docs['category'], test_docs['predicted']))
```

```
               precision    recall  f1-score   support

     business       0.95      0.91      0.93       510
entertainment       0.96      0.92      0.94       386
     politics       0.93      0.95      0.94       417
        sport       0.98      0.99      0.98       511
         tech       0.91      0.96      0.93       401

     accuracy                           0.95      2225
    macro avg       0.95      0.95      0.95      2225
 weighted avg       0.95      0.95      0.95      2225
```

Incluindo mais algumas stop_words, como 'said', 'also', 'one', 'two', 'would' que podem não agregar no processo de classificação da categoria da noticia.

```
In [567]: INDICE_NAME = 'category_index2'


analysis={
          "analyzer": {
            "analizer_text2": {
                  "char_filter": [
                    "replace_numbers"
                    ],
                  "tokenizer": "standard",
                  "filter": [
                      "lowercase",
                      "asciifolding",
                      "apostrophe",
                      "stop_custom",
                  ]
                }
              },
          "char_filter": {
             "replace_numbers": {
                    "type": "pattern_replace",
                    "pattern": "([0-9]+)",
                    "replacement": ""
             },
          },
          "filter": {
            "english_stop": {
              "type": "stop",
              "stopwords": "_english_"
            },
            "stop_custom": {
                "type": "stop",
                "stopwords": stop_words_en + ['said', 'also', 'one', 'two', 'woul
        },
    }
}

text_category_analizer = {
    "settings": {
        "analysis": analysis
      }
    }

if client.indices.exists(index=INDICE_NAME):
    client.indices.delete(index=INDICE_NAME)
client.indices.create(index=INDICE_NAME, **text_category_analizer)




INDEX_NAME = 'news_category_detection_data_2'

INDEX_MAPPING = {
        "settings": {
            "number_of_shards": 3,
            "analysis": analysis
```

```
    },
    "mappings": {
        "properties": {
            "category": {
                "type": "text",
                "analyzer": "analizer_text2",
                "fields": {
                    "raw": {
                        "type": "keyword"
                    }
                }
            },
            "title": {
                "type": "text",
                "analyzer": "analizer_text2",
                "fielddata": True,
                "fielddata_frequency_filter": {
                    "min": 0.01,
                    "min_segment_size": 10,
                    },
            },
            "content": {
                "type": "text",
                "analyzer": "analizer_text2",
                "fielddata": True,
                "fielddata_frequency_filter": {
                    "min": 0.01,
                    "min_segment_size": 10,
                }
            },
            "filename": {
                "type": "text",
                "fields": {
                    "raw": {
                        "type": "keyword"
                    }
                }
            },
        }
    }
}

if client.indices.exists(index=INDEX_NAME):
    client.indices.delete(index=INDEX_NAME)
client.indices.create(index=INDEX_NAME, **INDEX_MAPPING)
```

Out[567]:
```
ObjectApiResponse({'acknowledged': True, 'shards_acknowledged': True, 'index':
'news_category_detection_data_2'})
```

In [568]:
```
%%time
documents = gen_documents(df2_data)
actions = gen_index_actions(documents)

success, errors = bulk(client, actions)

client.indices.refresh(index=INDEX_NAME)
client.indices.flush(index=INDEX_NAME)
```

```
Wall time: 1.21 s
```

Out[568]:
```
ObjectApiResponse({'_shards': {'total': 6, 'successful': 3, 'failed': 0}})
```

In [569]:

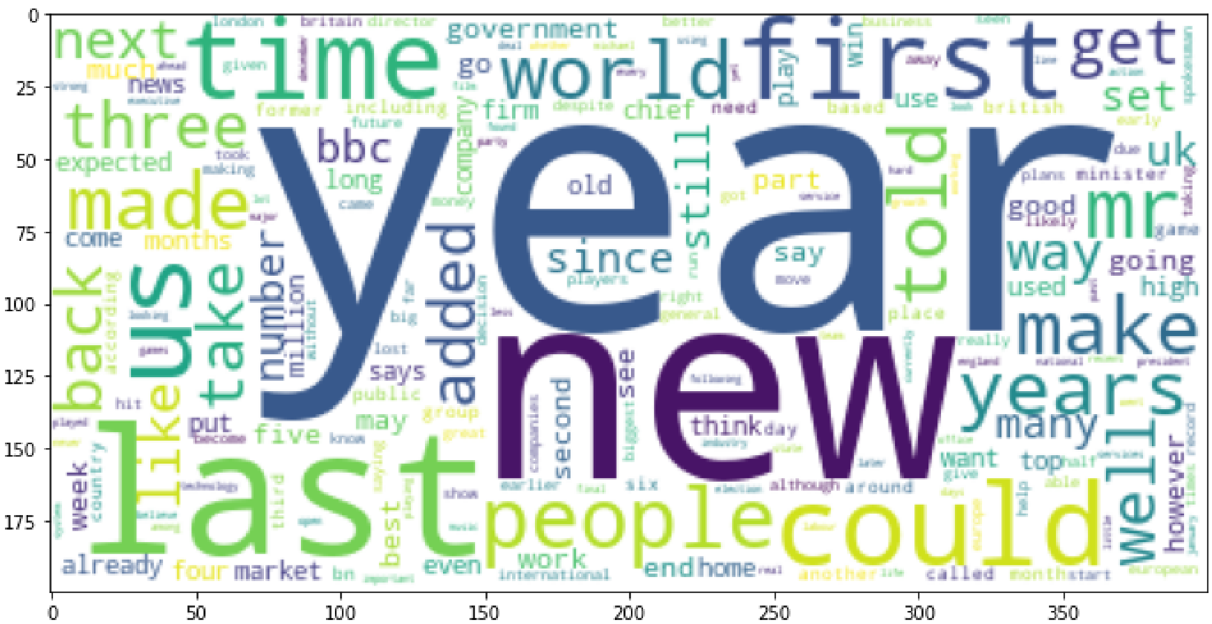
```python
QUERY = {
     'match_all': {}
}

AGG = {
    'grupos': {
        'terms': {
            'field': 'content',
            'size': 2000000
        },
    }
}
resp = client.search(index=INDEX_NAME, query=QUERY, aggregations=AGG, size=0)
frequencies = {}
buckets =resp['aggregations']['grupos']['buckets']
for bucket in buckets:
    frequencies[bucket['key']] = bucket['doc_count']

cloud = wordcloud.WordCloud(background_color='white')
cloud.generate_from_frequencies(frequencies)

fig, ax = plt.subplots(figsize=(16, 6))
ax.imshow(cloud)
```

Out[569]: <matplotlib.image.AxesImage at 0x1d8464164f0>



In [570]:
```
test_docs_resp = client.search(index=INDEX_NAME, size=2225, _source=['category'])
test_docs = pd.DataFrame({'_id': x['_id'],**x['_source']} for x in test_docs_resp
test_docs['predicted'] = test_docs['_id'].apply(classify_document2)
print(classification_report(test_docs['category'], test_docs['predicted']))
```

```
               precision    recall  f1-score   support

     business       0.96      0.92      0.93       510
entertainment       0.96      0.92      0.94       386
     politics       0.93      0.94      0.93       417
        sport       0.97      0.99      0.98       511
         tech       0.91      0.96      0.93       401

     accuracy                           0.95      2225
    macro avg       0.95      0.95      0.95      2225
 weighted avg       0.95      0.95      0.95      2225
```

In [ ]: Outro formato de consulta, com título e content

In [571]:
```
def classify_document3(doc_id, size=10):
    return classify_with_score3(
        [
            {
                '_index': INDEX_NAME,
                '_id': doc_id
            }
        ]
    )

def classify_with_score3(like, size=10):

    query = {
        'more_like_this': {
            'fields': ["title", "content"],
            'like':like,
            'min_term_freq': 1,
            'max_query_terms': 12,
            'minimum_should_match': -100,
        }
    }
    resp = client.search(index=INDEX_NAME, query=query, size=size)
    resp_df = pd.DataFrame(
            {
                '_id': x['_id'],
                '_score': x['_score'],
                **x['_source']
            } for x in resp['hits']['hits']
        )


    return resp_df.groupby('category').sum().sort_values('_score', ascending=Fals
```

In [572]:
```
test_docs_resp = client.search(index=INDEX_NAME, size=2225)
test_docs = pd.DataFrame({'_id': x['_id'],**x['_source']} for x in test_docs_resp
test_docs['predicted'] = test_docs['_id'].apply(classify_document3)
print(classification_report(test_docs['category'], test_docs['predicted']))
```

```
               precision    recall  f1-score   support

     business       0.95      0.92      0.93       510
entertainment       0.96      0.92      0.94       386
     politics       0.92      0.94      0.93       417
        sport       0.97      0.99      0.98       511
         tech       0.91      0.96      0.93       401

     accuracy                           0.95      2225
    macro avg       0.94      0.94      0.94      2225
 weighted avg       0.95      0.95      0.95      2225
```

Textos onde o classificador errou na predição, textos originais, sem analizador. Pode ser feito uma analise textual onde o o classificar não acertou para possível melhoria, ou quem sabe ser revisto a classificação do tópico.

In [575]:
```
for line in test_docs.index:
    if(test_docs['category'].iloc[line]!=test_docs['predicted'].iloc[line]):
        print(BOLD+"Predito:",test_docs['predicted'].iloc[line])
        print(BOLD+"Real:",test_docs['category'].iloc[line])
        print("------------------------------------------------")
        print(NORMAL+test_docs['content'].iloc[line])
```

```
Predito: tech
Real: business
------------------------------------------------
 The European Commission has written to the mobile phone operators Vodafone
 and T-Mobile to challenge "the high rates" they charge for international ro
aming.  In letters sent to the two companies, the Commission alleged the fir
ms were abusing their dominant market position in the German mobile phone ma
rket. It is the second time Vodafone has come under the Commission's scrutin
y. The UK operator is already appealing against allegations that its UK roam
ing rates are "unfair and excessive". Vodafone's response to the Commissio
n's letter was defiant. "We believe the roaming market is competitive and we
expect to resist the charges," said a Vodafone spokesman. "However we will n
eed time to examine the statement of objections in detail before we formally
respond."  The Commission's investigation into Vodafone and Deutsche Teleko
m's T-Mobile centres on the tariffs the two companies charge foreign mobile
 operators to access their networks when subscribers of those foreign operat
ors use their mobile phones in Germany.  The Commission believes these whole
sale prices are too high and that the excess is passed on to consumers. "The
Commission aims to ensure that European consumers are not overcharged when t
```

O primeiro dataset tem pouca informação textual para esse tipo de classificação, então diversos problemas foram verificados, como o retorno sem match de nenhum documento e a dificuldade em se classificar, que também pode ser vista pelo número alto ed categórias. O segundo data set tem mais texto e possui menos categórias, então o classificador tem alta taxa de acerto dos assuntos.