main

October 6, 2024

Objetivo: é a partir dos nomes e dos aliases de várias empresa, encontrar várias mencoes das mesmas em notícias e tentar …

1. grafo de palavras/pessoas/temas associadas [ver se é positivo / negativo o termo/pessoa]
2. relacao entre noticias e stock price
3. …

Trabalho tem de ter 3 partes:

1. project structure + data acquisition
2. exploratory data analysis and visualization
3. results & discussion

Fonte de Dados: arquivo.pt (https://github.com/arquivo/pwa-technologies/wiki/Arquivo.pt-API)

[3]:

```python
import pandas as pd
import requests
from bs4 import BeautifulSoup
```

sites dos quais vamos obter as noticias

[4]:

```python
# news from https://www.kadaza.pt

def news(txtFile = 'noticias.txt'):
    """
    grab the news websites from a text file
    """
    with open(txtFile, 'r') as file:
        links = file.read().splitlines()
    return ",".join(links)

#news()
```

como vão ser os api requests / decidir as empresas (PSI20) a analisar / fazer api requests in 3years groups

1 year to 3 years is long enough to smooth out short-term fluctuations and identify underlying trends. Charts with weekly or monthly intervals over these periods show developments over full economic/market cycles.

[5]:
```
def api_request(search, websites, date):
    """
    search: expression/word (what to look for)
    websites: comma separated websites (where to look for)
    date: list such as [20030101, 20031231] (when to look for)
    -
    returns the responde_items from arquivo.pt api
    """
    search = f"q=%22{search.replace(' ', '%20')}%22"
    websites = f"&siteSearch={websites}"
    date = f"&from={date[0]}&to={date[1]}"
    url = (
        f"https://arquivo.pt/textsearch?{search}{websites}{date}"
        "&fields=linkToArchive,linkToExtractedText,tstamp"
        "&maxItems=500&dedupValue=25&dedupField=url&prettyPrint=false&type=html"
        )
    json = requests.get(url).json()
    data = json["response_items"]
    if len(data) == 500:
        print(f"You might have lost some data: {search, date}")
    return data
```

[6]:
```
def datav1(companies):
    """
    this is the function where we choose the companies which will be in study
    -
    companies should be a dictionary
        {"company1": [aliases or other names the company is or was known by],
        "company2": [...]}
    -
    this data will be saved into a parquet file for future use and with already␣
 ↪api requests

    also this will do the api requests .... get this better
    """
    # CREATING DF WITH COMPANIES AND THEIR ALIASES
    companies_data = {"companies": [], "aliases": []}
    for company in companies.keys():
        companies_data["companies"].append(company)
        companies_data["aliases"].append(companies[company])
    df = pd.DataFrame(companies_data).set_index("companies")

    # SITES OF WHERE TO LOOK FOR NEWS
    websites = news()

    # INITIALIZAING API REQUESTS
    # groups of 3 years, from 2000 to 2020
    for cluster in range(2000, 2021, 3):
        api_cluster = [] #reset api_cluster for each cluster (group of 3 year)
        print(f"Processing cluster: {cluster}")
        print("Processing company:", end=" ")
        # iterate over each company
        for company_aliases in df["aliases"]:
            api_company = [] #reset api_company for each company
            print(f"{company_aliases[0]}", end = "; ")
            # iterate over each company's aliases
            for alias in company_aliases:
                # iterate over each cluter's year
                for year in range(cluster, cluster + 3):
                    api_aliasS1 = api_request(alias, websites,␣
 ↪[int(f"{year}0101"), int(f"{year}0630")])
                    api_aliasS2 = api_request(alias, websites,␣
 ↪[int(f"{year}0701"), int(f"{year}1231")])
                    api_company += api_aliasS1 + api_aliasS2
            # save company data
            api_cluster.append(api_company)

        # save cluster (group of 3 years) data
        df[f"api.{cluster}"] = api_cluster
        print(f"{cluster} OK.")

    # save all data
    df.to_parquet("data01.parquet")
    print("Finished.")
    return df

companies = {"Banco Comercial Português": ["Banco Comercial Português", "BCP"],
             "Galp Energia": ["Galp Energia", "GALP"],
             "EDP": ["EDP", "Energias de Portugal", "Electricidade de␣
 ↪Portugal"],
             "Sonae": ["Sonae", "SON"],
             "Mota-Engil": ["Mota-Engil", "EGL"]}
df01 = datav1(companies)
df01
```

```
Processing cluster: 2000
Processing company: Banco Comercial Português; Galp Energia; EDP; Sonae; Mota-
Engil; 2000 OK.
Processing cluster: 2003
Processing company: Banco Comercial Português; Galp Energia; EDP; Sonae; Mota-
Engil; 2003 OK.
Processing cluster: 2006
Processing company: Banco Comercial Português; Galp Energia; EDP; Sonae; Mota-
Engil; 2006 OK.
```

```
Processing cluster: 2009
Processing company: Banco Comercial Português; Galp Energia; EDP; Sonae; Mota-
Engil; 2009 OK.
Processing cluster: 2012
Processing company: Banco Comercial Português; Galp Energia; EDP; Sonae; Mota-
Engil; 2012 OK.
Processing cluster: 2015
Processing company: Banco Comercial Português; Galp Energia; EDP; Sonae; Mota-
Engil; 2015 OK.
Processing cluster: 2018
Processing company: Banco Comercial Português; Galp Energia; EDP; Sonae; Mota-
Engil; 2018 OK.
Finished.
```

[6]:

```
                                                                     aliases  \
companies
Banco Comercial Português                   [Banco Comercial Português, BCP]
Galp Energia                                            [Galp Energia, GALP]
EDP                        [EDP, Energias de Portugal, Electricidade de P…
Sonae                                                           [Sonae, SON]
Mota-Engil                                                 [Mota-Engil, EGL]

                                                                    api.2000  \
companies
Banco Comercial Português  [{'linkToArchive': 'https://arquivo.pt/wayback…
Galp Energia               [{'linkToArchive': 'https://arquivo.pt/wayback…
EDP                        [{'linkToArchive': 'https://arquivo.pt/wayback…
Sonae                      [{'linkToArchive': 'https://arquivo.pt/wayback…
Mota-Engil                 [{'linkToArchive': 'https://arquivo.pt/wayback…

                                                                    api.2003  \
companies
Banco Comercial Português  [{'linkToArchive': 'https://arquivo.pt/wayback…
Galp Energia               [{'linkToArchive': 'https://arquivo.pt/wayback…
EDP                        [{'linkToArchive': 'https://arquivo.pt/wayback…
Sonae                      [{'linkToArchive': 'https://arquivo.pt/wayback…
Mota-Engil                 [{'linkToArchive': 'https://arquivo.pt/wayback…

                                                                    api.2006  \
companies
Banco Comercial Português  [{'linkToArchive': 'https://arquivo.pt/wayback…
Galp Energia               [{'linkToArchive': 'https://arquivo.pt/wayback…
EDP                        [{'linkToArchive': 'https://arquivo.pt/wayback…
Sonae                      [{'linkToArchive': 'https://arquivo.pt/wayback…
Mota-Engil                 [{'linkToArchive': 'https://arquivo.pt/wayback…

                                                                    api.2009  \
companies
Banco Comercial Português  [{'linkToArchive': 'https://arquivo.pt/wayback…
Galp Energia               [{'linkToArchive': 'https://arquivo.pt/wayback…
EDP                        [{'linkToArchive': 'https://arquivo.pt/wayback…
Sonae                      [{'linkToArchive': 'https://arquivo.pt/wayback…
Mota-Engil                 [{'linkToArchive': 'https://arquivo.pt/wayback…

                                                                    api.2012  \
companies
Banco Comercial Português  [{'linkToArchive': 'https://arquivo.pt/wayback…
Galp Energia               [{'linkToArchive': 'https://arquivo.pt/wayback…
EDP                        [{'linkToArchive': 'https://arquivo.pt/wayback…
Sonae                      [{'linkToArchive': 'https://arquivo.pt/wayback…
Mota-Engil                 [{'linkToArchive': 'https://arquivo.pt/wayback…

                                                                    api.2015  \
companies
Banco Comercial Português  [{'linkToArchive': 'https://arquivo.pt/wayback…
Galp Energia               [{'linkToArchive': 'https://arquivo.pt/wayback…
EDP                        [{'linkToArchive': 'https://arquivo.pt/wayback…
Sonae                      [{'linkToArchive': 'https://arquivo.pt/wayback…
Mota-Engil                 [{'linkToArchive': 'https://arquivo.pt/wayback…

                                                                    api.2018
companies
Banco Comercial Português  [{'linkToArchive': 'https://arquivo.pt/wayback…
Galp Energia               [{'linkToArchive': 'https://arquivo.pt/wayback…
EDP                        [{'linkToArchive': 'https://arquivo.pt/wayback…
Sonae                      [{'linkToArchive': 'https://arquivo.pt/wayback…
Mota-Engil                 [{'linkToArchive': 'https://arquivo.pt/wayback…
```

```
[8]: df01.map(lambda x: len(x))
```

```
[8]:                            aliases  api.2000  api.2003  api.2006  api.2009  \
     companies
     Banco Comercial Português        2       153       241       183       561
     Galp Energia                     2       128       389       272       582
     EDP                              3       133       339       173       653
     Sonae                            2       192       435       279       502
     Mota-Engil                       2         4        83        60       195

                                api.2012  api.2015  api.2018
     companies
     Banco Comercial Português      1074      1430       954
     Galp Energia                   1156      1391       968
     EDP                            1232      1970      1096
     Sonae                          1215      1705      1196
```

```
Mota-Engil                                538         828         560
```

---

1 if the url is the same, check the content to see if its repeated

because we used `&dedupValue=25&dedupField=url` and different aliases, we might have repeated data, so it’s important to check for it

and also check for extractedText that doesn’t have our aliases

[ ]:

```
```