[Open In Colab](https://colab.research.google.com/gist/fernandoferreira-me/61e9e4a25060f95abbfdb8cb90aaaed9/question-rio-projeto-de-disciplina-de-tex)

Projeto de Disciplina de Processamento de Linguagem Natural com P

Bem-vindo ao projeto de disciplina de **Processamento de Linguagem Natural com Python**. Ao longo das últimas aulas vimos uma série possibilidades em trabalhar com textos. Para tal, usamos diversas bibliotecas, onde as que mais se destacaram foram NLTK, SPACY e GE

Esse notebook servirá de guia para a execução de uma análise de tópicos completa, usando o algoritmo de LDA e recursos para interpreta "Mercado" extraídas da Folha de S. Paulo no ano de 2016. Complete a análise com os códigos que achar pertinente e responda as questõ

O Notebook

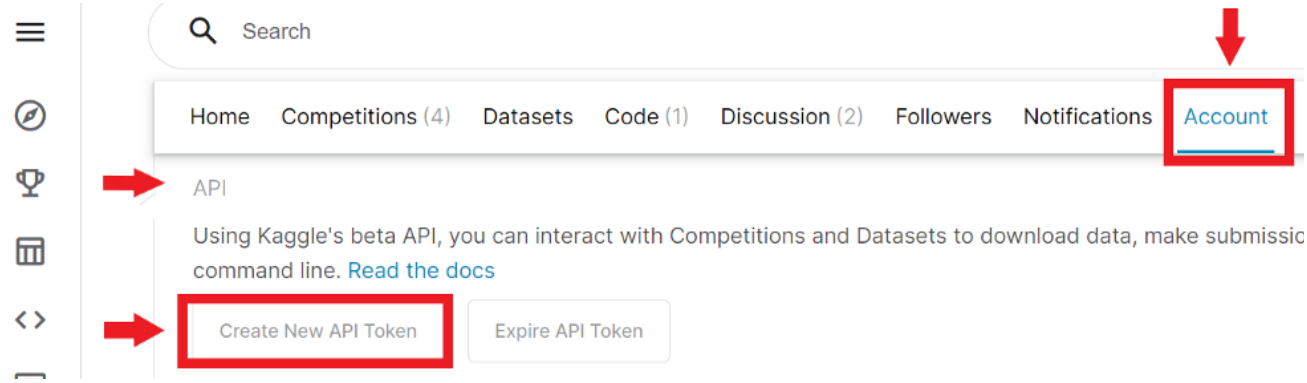
Nesse notebook, você será guiado pela análise de **Extração de Tópicos**. As seguintes tarefas serão realizadas

1. Download dos dados provenientes do kaggle
2. Seleção dos dados relevantes para a nossa análise
3. Instalação das principais ferramentas e importação de módulos
4. Pré-processamento usando NLTK
5. Pré-processamento usando Spacy
6. Análise de tópicos usando LDA
7. Análise de NER usando Spacy
8. Visualização dos tópicos usando tokens e entidades.

Instruções para baixar os dados

Para baixar os dados será necessário o uso do gerenciador de downloads da Kaggle. A Kaggle, uma subsidiária do grupo Alphabet (Googl profissionais de aprendizado de máquina.

Para utilizar o gerenciador, será necessário criar uma conta no site Kaggle.com. Com a conta criada, obtenha um token de acesso, no form



Em posse do token (baixe para seu computador), execute a células da próxima seção para acessar os dados de interesse e baixá-los.

1. Qual o endereço do seu notebook (colab) executado? Use o botão de compartilhamento do colab para obter uma url

O notebook foi executado localmente em um notebook mantido pelo Anaconda, entretanto disponibilizei o código, pdf, html e requirements

https://github.com/Herikc2/Processamento-de-linguagem-natural-com-Python (https://github.com/Herikc2/Processamento-de-linguagem-na

Baixe os dados

Instale o gerenciador kaggle no ambiente do Colab e faça o upload do arquivo kaggle.json

In [1]:
```
# Instalando depedências para importar notebook do kaggle
!pip install -q kaggle
!pip install -q opendatasets
```

```
[notice] A new release of pip is available: 23.0.1 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip

[notice] A new release of pip is available: 23.0.1 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

4. Em qual célula está o download dos dados diretamente do kaggle?

Realizei o download utilizando o opendatasets. O kaggle.json esta no mesmo diretório no notebook.

In [2]:
```
import opendatasets as od
import shutil
od.download("https://www.kaggle.com/datasets/marlesson/news-of-the-site-folhauol")

source_folder = 'news-of-the-site-folhauol'
destination_folder = '../data/news-of-the-site-folhauol'

shutil.move(source_folder, destination_folder)
```

```
Downloading news-of-the-site-folhauol.zip to .\news-of-the-site-folhauol

100%|████████████████████████████████████████████████████████████████████████████████| 187M/187M [00:21<00:00, 9.
```

Out[2]:
```
'../data/news-of-the-site-folhauol\\news-of-the-site-folhauol'
```

Criar o DataFrame com os dados lidos diretamente da plataforma Kaggle

In [3]:
```
# Importdando os dados e bibliotecas de manipulação dos dados
import pandas as pd
from tqdm.auto import tqdm
tqdm.pandas()

df = pd.read_csv(r"..\data\news-of-the-site-folhauol\articles.csv")
```

```
C:\Users\herik\anaconda3\lib\site-packages\numpy\_distributor_init.py:30: UserWarning: loaded more than 1 DLL fr
C:\Users\herik\anaconda3\lib\site-packages\numpy\.libs\libopenblas.4SP5SUA7CBGXUEOC35YP2ASOICYYEQZZ.gfortran-win
C:\Users\herik\anaconda3\lib\site-packages\numpy\.libs\libopenblas64__v0.3.21-gcc_10_3_0.dll
  warnings.warn("loaded more than 1 DLL from .libs:"
```

In [4]:
```
df.head()
```

Out[4]:

	title	text	date	category	subcategory	
0	Lula diz que está 'lascado', mas que ainda tem...	Com a possibilidade de uma condenação impedir ...	2017-09-10	poder	NaN	http://w
1	'Decidi ser escrava das mulheres que sofrem', ...	Para Oumou Sangaré, cantora e ativista malines...	2017-09-10	ilustrada	NaN	http:/
2	Três reportagens da Folha ganham Prêmio Petrob...	Três reportagens da Folha foram vencedoras do ...	2017-09-10	poder	NaN	http://w
3	Filme 'Star Wars: Os Últimos Jedi' ganha trail...	A Disney divulgou na noite desta segunda-feira...	2017-09-10	ilustrada	NaN	http:/
4	CBSS inicia acordos com fintechs e quer 30% do...	O CBSS, banco da holding Elopar dos sócios Bra...	2017-09-10	mercado	NaN	http://w

Atualizar o SPACY e instalar os modelos pt_core_news_lg

3. Em qual célula está o código que atualiza o spacy e instala o pacote pt_core_news_lg?

Atualizando o spacy para a ultima versão disponível e instalando em moto quiet. Realizando download local do pt_core_news_lg.

In [5]:
```
!pip install -q -U spacy

import spacy
from spacy.lang.pt.stop_words import STOP_WORDS

# Baixando pacote de NLP em português
spacy.cli.download("pt_core_news_lg")
```

```
[notice] A new release of pip is available: 23.0.1 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip

✓ Download and installation successful
You can now load the package via spacy.load('pt_core_news_lg')
```

Instalar os datasets `stopwords`, `punkt` e `rslp` do nltk

In [6]:
```
# Baixando ferramentas de NLP do NLTK

import nltk

nltk.download("stopwords")
nltk.download("punkt")
nltk.download("rslp")

from nltk.corpus import stopwords
from nltk.tokenize import punkt
from nltk.stem import rslp
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\herik\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\herik\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package rslp to
[nltk_data]     C:\Users\herik\AppData\Roaming\nltk_data...
[nltk_data]   Package rslp is already up-to-date!
```

Carregar os módulos usados ao longo desse notebook

In [7]:
```
!pip install pyldavis==3.4.1

import warnings
warnings.filterwarnings('ignore')

from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation as LDA
import numpy as np

import pyLDAvis
import pyLDAvis.lda_model
pyLDAvis.enable_notebook()

from wordcloud import WordCloud

import seaborn as sns
import matplotlib.pyplot as plt
from itertools import chain

from typing import List, Set, Any

SEED = 123
debug = False
```

```
Requirement already satisfied: pyldavis==3.4.1 in c:\users\herik\anaconda3\lib\site-packages (3.4.1)
Requirement already satisfied: gensim in c:\users\herik\appdata\roaming\python\python39\site-packages (from pyld
Requirement already satisfied: joblib>=1.2.0 in c:\users\herik\anaconda3\lib\site-packages (from pyldavis==3.4.1
Requirement already satisfied: scipy in c:\users\herik\appdata\roaming\python\python39\site-packages (from pylda
Requirement already satisfied: funcy in c:\users\herik\appdata\roaming\python\python39\site-packages (from pylda
Requirement already satisfied: setuptools in c:\users\herik\anaconda3\lib\site-packages (from pyldavis==3.4.1) (
Requirement already satisfied: pandas>=2.0.0 in c:\users\herik\anaconda3\lib\site-packages (from pyldavis==3.4.1
Requirement already satisfied: scikit-learn>=1.0.0 in c:\users\herik\anaconda3\lib\site-packages (from pyldavis=
Requirement already satisfied: numexpr in c:\users\herik\anaconda3\lib\site-packages (from pyldavis==3.4.1) (2.8
Requirement already satisfied: jinja2 in c:\users\herik\anaconda3\lib\site-packages (from pyldavis==3.4.1) (2.11
Requirement already satisfied: numpy>=1.24.2 in c:\users\herik\anaconda3\lib\site-packages (from pyldavis==3.4.1
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\herik\anaconda3\lib\site-packages (from pandas
Requirement already satisfied: pytz>=2020.1 in c:\users\herik\anaconda3\lib\site-packages (from pandas>=2.0.0->p
Requirement already satisfied: tzdata>=2022.1 in c:\users\herik\anaconda3\lib\site-packages (from pandas>=2.0.0-
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\herik\anaconda3\lib\site-packages (from scikit-l
Requirement already satisfied: six>=1.5.0 in c:\users\herik\anaconda3\lib\site-packages (from gensim->pyldavis==
Requirement already satisfied: smart-open>=1.8.1 in c:\users\herik\anaconda3\lib\site-packages (from gensim->pyl
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\herik\anaconda3\lib\site-packages (from jinja2->pyld
Requirement already satisfied: packaging in c:\users\herik\anaconda3\lib\site-packages (from numexpr->pyldavis==
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\herik\anaconda3\lib\site-packages (from pack


[notice] A new release of pip is available: 23.0.1 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
C:\Users\herik\anaconda3\lib\site-packages\seaborn\rcmod.py:82: DeprecationWarning: distutils Version classes ar
  if LooseVersion(mpl.__version__) >= "3.0":
C:\Users\herik\anaconda3\lib\site-packages\setuptools\_distutils\version.py:351: DeprecationWarning: distutils V
  other = LooseVersion(other)
```

Filtrando os dados para utilizar apenas as notícias do ano de 2016 e d

Filtre os dados do DataFrame df e crie um DataFrame news_2016 que contenha apenas notícias de **2016** e da categoria **mercado**.

5. Em qual célula está a criação do dataframe news_2016 (com examente 7943 notícias)?

Caso debug seja False irá realizar os filtros solicitados no projeto, caso True irá executar de forma reduzida para encurtar.

In [8]:
```
df['date'] = pd.to_datetime(df.date)

# Se False irá carregar os dados conforme solicitação do projeto, caso contrário irá carregar uma sample menor
if debug == False:
  # Create a dataframe named news_2016
  news_2016 = df.loc[(df['date'].dt.year == 2016) & (df['category'] == 'mercado')].reset_index(drop = True)
else:
  # When debugin will filter just by 2016 and first month to execute faster
  news_2016 = df.loc[(df['date'].dt.year == 2016) & (df['date'].dt.month == 1) & (df['category'] == 'mercado')].
```

In [9]:
```
news_2016.shape
```

Out[9]:
```
(7943, 6)
```

In [10]:
```
news_2016.head()
```

Out[10]:

	title	text	date	category	subcategory	
0	Fazendeira cria própria rede de banda larga e ...	"Sou apenas a mulher de um fazendeiro", diz Ch...	2016-12-31	mercado	NaN	http://ww
1	Alteração na cobrança do ICMS eleva conta de c...	A conta do celular pós-pago ou controle ficará...	2016-12-31	mercado	NaN	http://ww
2	Ajustes sobre servidores públicos emperram nos...	A maior parte dos projetos de ajuste das conta...	2016-12-31	mercado	NaN	http://ww
3	Inventor da internet das coisas ataca mitos so...	Desde as primeiras décadas do século 19 se diz...	2016-12-31	mercado	NaN	http://ww
4	Livro analisa empresas de crescimento exponenc...	O Cifras & Letras seleciona semanalmente lança...	2016-12-31	mercado	NaN	http://ww

NLTK Tokenizer and Stemmer

Crie uma coluna no dataframe `news_2016` contendo os tokens para cada um dos textos. Os tokens devem estar representados pelo radic
função `tokenize` .

2. Em qual célula está o código que realiza o download dos pacotes necessários para tokenização e stemming usando nltk?

Importação dos pacotes utilizados para tokenização e stemming.

In [11]:
```
from nltk.tokenize import word_tokenize
from nltk.stem import RSLPStemmer
```

6. Em qual célula está a função que tokeniza e realiza o stemming dos textos usando funções do nltk?

Abaixo esta realizando a função de tokenização, a coluna gerada não é utilizada para outros fins durante o código.

In [12]:
```
def tokenize(text: str) -> List:
    """
      Function for tokenizing using `nltk.tokenize.word_tokenize`

      Returns:
        - A list of stemmed tokens (`nltk.stem.RSLPStemmer`)
        IMPORTANT: Only tokens with alphabetic
                   characters will be returned.
    """
    st = RSLPStemmer()

    token_text = word_tokenize(text, language = 'portuguese')

    return [st.stem(word) for word in token_text if word.isalnum()]

news_2016.loc[:, 'nltk_tokens'] = news_2016.text.progress_map(tokenize)
```

100% 7943/7943 [02:12<00:00, 49.95it/s]

In [13]:
```
news_2016.head()
```

Out[13]:

	title	text	date	category	subcategory	
0	Fazendeira cria própria rede de banda larga e ...	"Sou apenas a mulher de um fazendeiro", diz Ch...	2016-12-31	mercado	NaN	http://www
1	Alteração na cobrança do ICMS eleva conta de c...	A conta do celular pós-pago ou controle ficará...	2016-12-31	mercado	NaN	http://www
2	Ajustes sobre servidores públicos emperram nos...	A maior parte dos projetos de ajuste das conta...	2016-12-31	mercado	NaN	http://www
3	Inventor da internet das coisas ataca mitos so...	Desde as primeiras décadas do século 19 se diz...	2016-12-31	mercado	NaN	http://www
4	Livro analisa empresas de crescimento exponenc...	O Cifras & Letras seleciona semanalmente lança...	2016-12-31	mercado	NaN	http://www

Criar uma documento SPACY para cada texto do dataset

Crie uma coluna `spacy_doc` que contenha os objetos spacy para cada texto do dataset de interesse. Para tal, carregue os modelos `pt_` demorar alguns minutos...)

9. Em qual célula o modelo pt_core_news_lg está sendo carregado? Todos os textos do dataframe precisam ser analisados usan feito?

Abaixo esta sendo carregado o pt_core_news_lg e gerado os documentos para cada linha do dataframe.

In [14]:
```
# Gerando coluna de Document usando o spacy
nlp_pt = spacy.load("pt_core_news_lg")

news_2016.loc[:, 'spacy_doc'] = news_2016['text'].apply(nlp_pt)
```

In [15]:
```
news_2016.head()
```

Out[15]:

	title	text	date	category	subcategory	
0	Fazendeira cria própria rede de banda larga e ...	"Sou apenas a mulher de um fazendeiro", diz Ch...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/
1	Alteração na cobrança do ICMS eleva conta de c...	A conta do celular pós-pago ou controle ficará...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/
2	Ajustes sobre servidores públicos emperram nos...	A maior parte dos projetos de ajuste das conta...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/
3	Inventor da internet das coisas ataca mitos so...	Desde as primeiras décadas do século 19 se diz...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/
4	Livro analisa empresas de crescimento exponenc...	O Cifras & Letras seleciona semanalmente lança...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/

Realize a Lematização usando SPACY

O modelo NLP do spacy oferece a possiblidade de lematizar textos em português (o que não acontece com a biblioteca NLTK). Iremos cria nosso dataset. Para tal, iremos retirar as stopwords, usando uma função que junta stopwords provenientes do NLTK e do Spacy. Essa lista não precisa mexer).

Já a função filter retorna True caso o token seja composto por caracters alfabéticos, não estiver dentro da lista de stopwords e o lemma re `"em a" e "ano"` .

Crie uma coluna chamada `spacy_lemma` para armazenar o resultado desse pré-processamento.

In [16]:
```
# Stopwords manuais
additional_stop_words = ['de', 'a', 'o', 'que', 'e', 'do', 'da', 'em', 'um', 'para', 'é', 'com', 'nã
```

7. Em qual célula está a função que realiza a lematização usando o spacy?

In [17]:
```
def stopwords() -> Set:
    """
    Return complete list of stopwords
    """
    return set(list(nltk.corpus.stopwords.words("portuguese")) + list(STOP_WORDS) + list(additional_stop_words))
    #return set(list(nltk.corpus.stopwords.words("portuguese")) + list(STOP_WORDS))

complete_stopwords = stopwords()

def filter(w: spacy.lang.pt.Portuguese) -> bool:
    """
    Filter stopwords and undesired tokens
    """
    undesired_tokens = ["o", "em", "em o", "em a", "ano"]

    w = w.lemma_.lower().strip()

    # Removendo caracteres não alfanumeriscos, stopwords e lista de não desejados
    if w.isalpha() and w not in complete_stopwords and w not in undesired_tokens:
        return True
    else:
        return False


def lemma(doc: spacy.lang.pt.Portuguese) -> List[str]:
    """
    Apply spacy lemmatization on the tokens of a text

    Returns:
       - a list representing the standardized (with lemmatisation) vocabulary
    """

    return [str(token.lemma_) for token in doc if filter(token)]

news_2016.loc[:, 'spacy_lemma'] = news_2016.spacy_doc.progress_map(lemma)
```

```
100%                                                  7943/7943 [00:03<00:00, 2448.99it/s]
```

In [18]:
```
news_2016.head()
```

Out[18]:

	title	text	date	category	subcategory	link	
0	Fazendeira cria própria rede de banda larga e ...	"Sou apenas a mulher de um fazendeiro", diz Ch...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	
1	Alteração na cobrança do ICMS eleva conta de c...	A conta do celular pós-pago ou controle ficará...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	
2	Ajustes sobre servidores públicos emperram nos...	A maior parte dos projetos de ajuste das conta...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	
3	Inventor da internet das coisas ataca mitos so...	Desde as primeiras décadas do século 19 se diz...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	
4	Livro analisa empresas de crescimento exponenc...	O Cifras & Letras seleciona semanalmente lança...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[c

8. Baseado nos resultados qual a diferença entre stemming e lematização, qual a diferença entre os dois procedimentos? Escolh

Stemming: Mantem somente a raiz.

Lematização: Reduz para palavra base.

Exemplos:

Original	Stemmer	Lemma
amigos	amig	amigo
amigas	amig	amigo
amizade	amizad	amizade
carreira	carr	carreira
carreiras	carr	carreira

Reconhecimento de entidades nomeadas

Crie uma coluna `spacy_ner` que armazene todas as organizações (APENAS organizações) que estão contidas no texto.

10. Indique a célula onde as entidades dos textos foram extraídas. Estamos interessados apenas nas organizações.

Abaixo esta sendo gerado as entidades filtrando por organização conforme desejado, a organização é extraida diretamente do spacy docu

In [19]:
```
def NER(doc: spacy.lang.pt.Portuguese):
    """
      Return the list of organizations for a SPACY document
    """

    target_label = 'ORG'

    entities = [(token.text, token.label_) for token in doc.ents]

    desired_entities = [entity[0] for entity in entities if entity[1] == target_label]

    return desired_entities

news_2016.loc[:, 'spacy_ner'] = news_2016.spacy_doc.progress_map(NER)
```

```
100%                                                  7943/7943 [00:00<00:00, 13158.11it/s]
```

In [20]:
```
news_2016.head()
```

Out[20]:

	title	text	date	category	subcategory	link	nltk_to
0	Fazendeira cria própria rede de banda larga e ...	"Sou apenas a mulher de um fazendeiro", diz Ch...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[sou, apen, a, de, um, fazenc
1	Alteração na cobrança do ICMS eleva conta de c...	A conta do celular pós-pago ou controle ficará...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[a, cont, do, celu control, fic, mai
2	Ajustes sobre servidores públicos emperram nos...	A maior parte dos projetos de ajuste das conta...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[a, mai, par projet, de, ajus c
3	Inventor da internet das coisas ataca mitos so...	Desde as primeiras décadas do século 19 se diz...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[desd, as, prim do, sécul, 19, se
4	Livro analisa empresas de crescimento exponenc...	O Cifras & Letras seleciona semanalmente lança...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[o, cifr, letr, s seman, lanç, na

Salvando/Carregando Spacy Data

Salvando o dataframe localmente para não precisar executar todas as celulas novamente.

In [21]:
```
# Carrega os dados localmente ou salva
if not 'news_2016' in locals():
    news_2016 = pd.read_pickle("../data/news_2016")
else:
    news_2016.to_pickle("../data/news_2016")
```

In [22]:
```
news_2016.head(3)
```

Out[22]:

	title	text	date	category	subcategory	link	nltk_
0	Fazendeira cria própria rede de banda larga e ...	"Sou apenas a mulher de um fazendeiro", diz Ch...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[sou, apen, a de, um, faze
1	Alteração na cobrança do ICMS eleva conta de c...	A conta do celular pós-pago ou controle ficará...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[a, cont, do, ce control, fic, m
2	Ajustes sobre servidores públicos emperram nos...	A maior parte dos projetos de ajuste das conta...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[a, mai, p projet, de, aj

Bag-of-Words

Crie uma coluna `tfidf` no dataframe `news_2016` . Use a coluna `spacy_lemma` como base para cálculo do TFIDF. O número máximo d que ter aparecido pelo menos 10 vezes (`min_df`) nos documentos.

In [23]:
```
# Convertendo coluna de lemmas para coluna
doc_tokens = news_2016.spacy_lemma.values.tolist()
```

In [24]:
```
# Gerando TF para comparações

# Criando corpus com todos os lemmas separados por espaço
corpus = [' '.join(tokens) for tokens in doc_tokens]

tf_vectorizer = CountVectorizer(max_features = 5000, min_df = 10, lowercase = True)

tf_matrix = tf_vectorizer.fit_transform(corpus)
```

13. Indique a célula onde está a função que cria o vetor de TF-IDF para cada texto.

Abaixo esta sendo criada a classe onde é construido o TFIDF, além disso é armazenado a matrix resultante do treinamento em uma veriav

In [25]:
```
class Vectorizer:
    def __init__(self, doc_tokens: List):
        self.doc_tokens = doc_tokens
        self.tfidf = None
        self.tfidf_matrix = None

    def vectorizer(self):
        """
        Convert a list of tokens to tfidf vector
        Returns the tfidf vector and attribute it to self.tfidf
        """

        # Gerando corpus a partir do documento alvo
        corpus = [' '.join(tokens) for tokens in self.doc_tokens]

        # Utilizando mesmo parâmetros do TF
        self.tfidf = TfidfVectorizer(**tf_vectorizer.get_params())

        # Treinando TFIDF
        self.tfidf_matrix = self.tfidf.fit_transform(corpus)

    def __call__(self):
        if self.tfidf is None:
            self.vectorizer()
        return self.tfidf

vectorizer = Vectorizer(doc_tokens)

def tokens2tfidf(tokens):
    text = ' '.join(tokens)
    array = vectorizer().transform([text]).toarray()[0]
    return array


news_2016.loc[:, 'tfidf'] = news_2016.spacy_lemma.progress_map(tokens2tfidf)
```

```
100%                                                  7943/7943 [00:13<00:00, 903.89it/s]
```

In [26]:
```
news_2016.head()
```

Out[26]:

	title	text	date	category	subcategory	link	nltk_tokens	
0	Fazendeira cria própria rede de banda larga e ...	"Sou apenas a mulher de um fazendeiro", diz Ch...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[sou, apen, a, mulh, de, um, fazend, diz, chri...	(", So mu
1	Alteração na cobrança do ICMS eleva conta de c...	A conta do celular pós-pago ou controle ficará...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[a, cont, do, celul, ou, control, fic, mais, c...	(celul c
2	Ajustes sobre servidores públicos emperram nos...	A maior parte dos projetos de ajuste das conta...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[a, mai, part, do, projet, de, ajust, da, cont...	(A, dos,
3	Inventor da internet das coisas ataca mitos so...	Desde as primeiras décadas do século 19 se diz...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[desd, as, prim, déc, do, sécul, 19, se, diz, ...	primeir do,
4	Livro analisa empresas de crescimento exponenc...	O Cifras & Letras seleciona semanalmente lança...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[o, cifr, letr, selec, seman, lanç, na, áre, d...	(O, Cifr sem

Extração de Tópicos

Realize a extração de 9 tópicos usando a implementação do sklearn do algoritmo Latent Dirichlet Allocation. Como parâmetros, você irá us
demorar) e o `random_seed` igual a `SEED` que foi setado no início do notebook

In [27]:
```
# Definir o número de tópicos desejado
N_TOKENS = 9

# Get the feature names from the TfidfVectorizer object
feature_names = vectorizer().get_feature_names_out()
```

In [28]:
```
corpus = np.array(news_2016.tfidf.tolist())

# Cria modelo do LDA baseado somente no corpus
lda_model_corpus = LDA(n_components=N_TOKENS, random_state=SEED)
lda_model_corpus.fit(corpus)

# Cria mapa por topico
topic_words = {}
for i, topic in enumerate(lda_model_corpus.components_):
    word_idx = topic.argsort()[::-1][:10]
    topic_words["Tópico %d" % (i+1)] = [feature_names[w] for w in word_idx]

# Imprime o mapa por topico
for topic, words in topic_words.items():
    print(topic + ":")
    print(words)
    print()
```

```
Tópico 1:
['empresa', 'milhão', 'companhia', 'negócio', 'serviço', 'brasil', 'venda', 'bilhão', 'mercado', 'outro']

Tópico 2:
['imóvel', 'poupança', 'caixa', 'grécia', 'fgts', 'greve', 'imobiliário', 'resgate', 'grego', 'atendimento']

Tópico 3:
['editora', 'pág', 'voo', 'crédito', 'autor', 'aéreo', 'cartão', 'gol', 'inadimplência', 'aeronave']

Tópico 4:
['petrobras', 'petróleo', 'bilhão', 'companhia', 'empresa', 'estatal', 'gás', 'venda', 'energia', 'milhão']

Tópico 5:
['contribuinte', 'declaração', 'ficha', 'restituição', 'receita', 'energia', 'aneel', 'rendimento', 'bandeira',

Tópico 6:
['índice', 'queda', 'dólar', 'banco', 'mercado', 'alta', 'juro', 'petróleo', 'subir', 'preço']

Tópico 7:
['pokémon', 'go', 'arena', 'espm', 'nintendo', 'publicidade', 'jogador', 'jogo', 'marketing', 'folha']

Tópico 8:
['governo', 'bilhão', 'presidente', 'proposta', 'público', 'ministro', 'temer', 'medida', 'afirmar', 'dívida']

Tópico 9:
['aposentadoria', 'benefício', 'idade', 'inss', 'contribuição', 'segurado', 'aposentar', 'perícia', 'note', 'sam
```

In [29]:
```
# Cria modelo do LDA baseado no TF gerado anteriormente
lda_model_tf = LDA(n_components=N_TOKENS, random_state=SEED)
lda_model_tf.fit(tf_matrix)

# Cria mapa por topico
topic_words = {}
for i, topic in enumerate(lda_model_tf.components_):
    word_idx = topic.argsort()[::-1][:10]
    topic_words["Tópico %d" % (i+1)] = [feature_names[w] for w in word_idx]

# Imprime o mapa por topico
for topic, words in topic_words.items():
    print(topic + ":")
    print(words)
    print()
```

```
Tópico 1:
['empresa', 'veículo', 'carro', 'milhão', 'companhia', 'justiça', 'afirmar', 'decisão', 'acordo', 'caso']

Tópico 2:
['país', 'brasil', 'petróleo', 'produção', 'preço', 'mercado', 'acordo', 'empresa', 'china', 'brasileiro']

Tópico 3:
['bilhão', 'banco', 'crédito', 'milhão', 'trimestre', 'empresa', 'taxa', 'juro', 'dívida', 'financeiro']

Tópico 4:
['trabalho', 'pessoa', 'país', 'emprego', 'economia', 'outro', 'crise', 'mercado', 'algum', 'renda']

Tópico 5:
['energia', 'pagar', 'informar', 'receber', 'receita', 'dia', 'dever', 'imposto', 'pagamento', 'declaração']

Tópico 6:
['mercado', 'queda', 'índice', 'alta', 'dólar', 'banco', 'juro', 'subir', 'cair', 'ação']

Tópico 7:
['empresa', 'produto', 'serviço', 'outro', 'venda', 'mercado', 'rede', 'negócio', 'milhão', 'afirmar']

Tópico 8:
['empresa', 'governo', 'bilhão', 'petrobras', 'investimento', 'presidente', 'companhia', 'estatal', 'projeto', '

Tópico 9:
['governo', 'público', 'presidente', 'proposta', 'medida', 'temer', 'afirmar', 'gasto', 'ministro', 'fiscal']
```

14. Indique a célula onde estão sendo extraídos os tópicos usando o algoritmo de LDA.

Abaixo é gerado o LDA utilizando a matriz resultante das transformações do TFIDF, além disso é realizado a impressão das 10 palavras cc

In [30]:
```
# Cria modelo do LDA baseado no TFIDF gerado anteriormente
lda_model_tfidf = LDA(n_components=N_TOKENS, random_state=SEED)
lda_model_tfidf.fit(vectorizer.tfidf_matrix)

# Cria mapa por topico
topic_words = {}
for i, topic in enumerate(lda_model_tfidf.components_):
    word_idx = topic.argsort()[::-1][:10]
    topic_words["Tópico %d" % (i+1)] = [feature_names[w] for w in word_idx]

# Imprime o mapa por topico
for topic, words in topic_words.items():
    print(topic + ":")
    print(words)
    print()
```

```
Tópico 1:
['empresa', 'milhão', 'companhia', 'negócio', 'serviço', 'brasil', 'venda', 'bilhão', 'mercado', 'outro']

Tópico 2:
['imóvel', 'poupança', 'caixa', 'grécia', 'fgts', 'greve', 'imobiliário', 'resgate', 'grego', 'atendimento']

Tópico 3:
['editora', 'pág', 'voo', 'crédito', 'autor', 'aéreo', 'cartão', 'gol', 'inadimplência', 'aeronave']

Tópico 4:
['petrobras', 'petróleo', 'bilhão', 'companhia', 'empresa', 'estatal', 'gás', 'venda', 'energia', 'milhão']

Tópico 5:
['contribuinte', 'declaração', 'ficha', 'restituição', 'receita', 'energia', 'aneel', 'rendimento', 'bandeira',

Tópico 6:
['índice', 'queda', 'dólar', 'banco', 'mercado', 'alta', 'juro', 'petróleo', 'subir', 'preço']
```

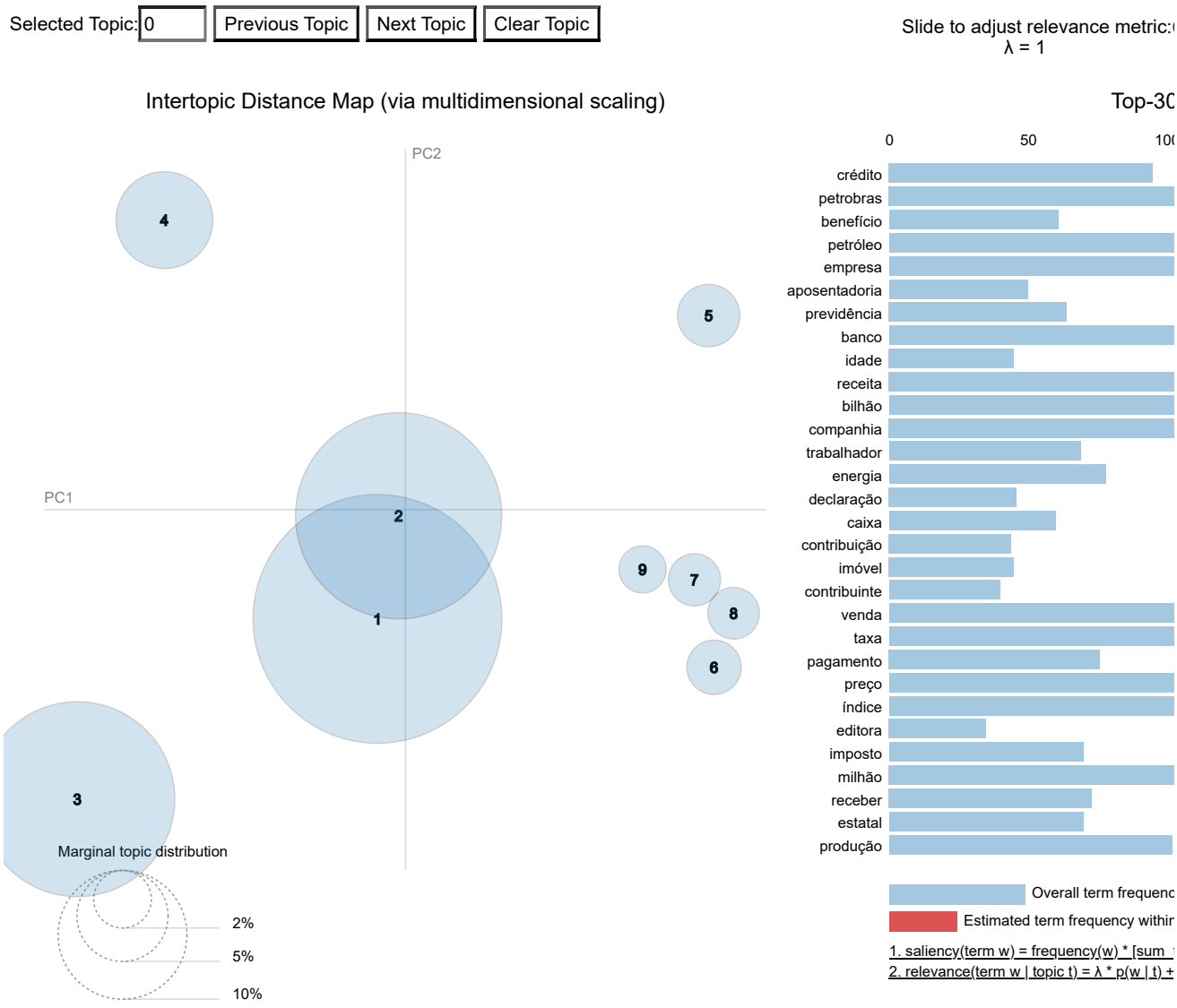
Visualize os tópicos usando a ferramenta pyLDAVis

15. Indique a célula onde a visualização LDAVis está criada.

Impressão do pyLDAvis utilizado pelo modelo de LDA TFIDF.

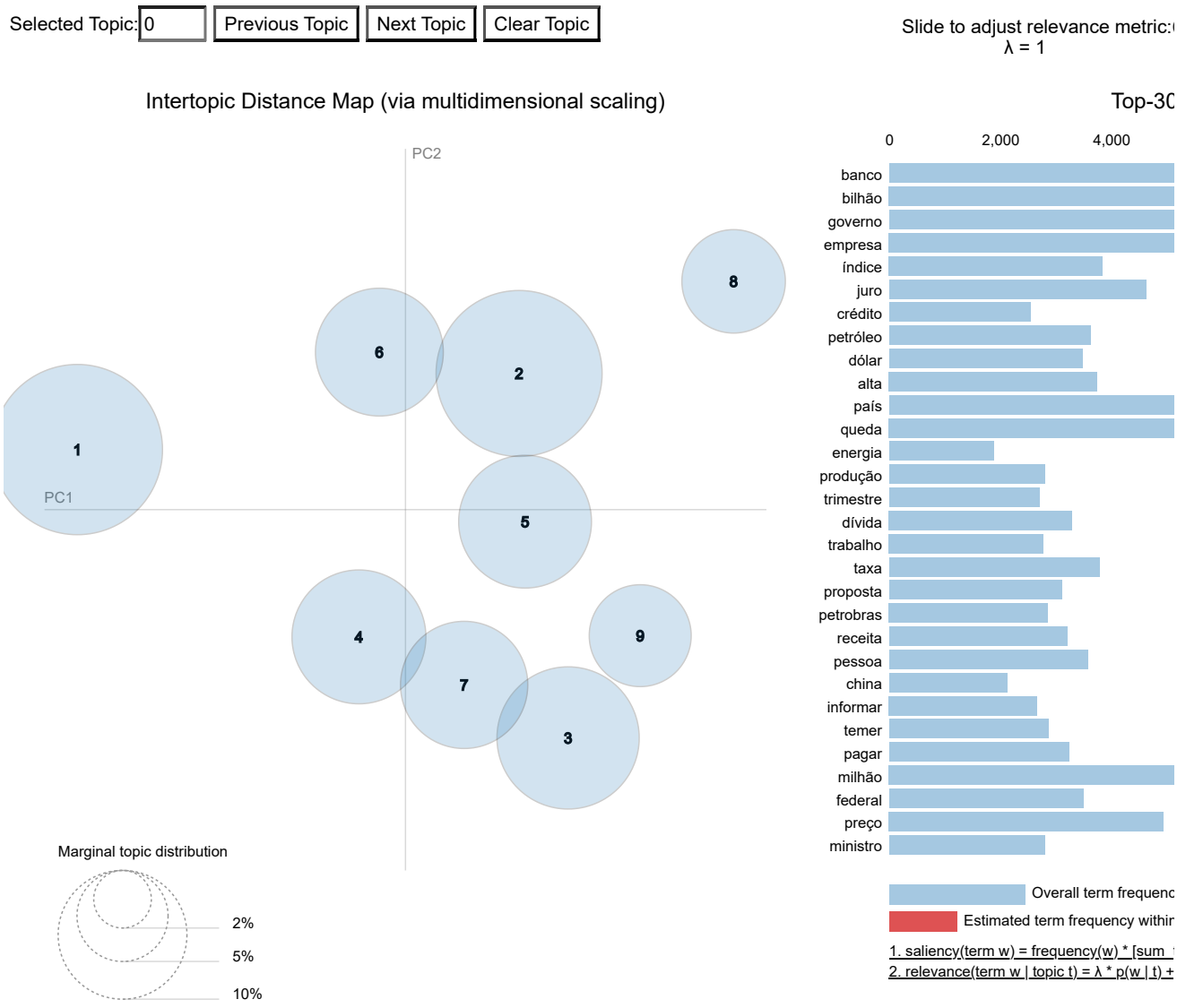
In [31]:
```
# pyLDAVis para o modelo de TFIDF
pyLDAvis.lda_model.prepare(lda_model_tfidf, vectorizer.tfidf_matrix, vectorizer())
```

Out[31]:



In [32]:
```
# pyLDAVis para o modelo de TF
pyLDAvis.lda_model.prepare(lda_model_tf, tf_matrix, tf_vectorizer)
```

Out[32]:



Atribua a cada text, um (e apenas um) tópic.

Crie uma coluna `topic` onde o valor é exatamente o tópico que melhor caracteriza o documento de acordo com o algoritmo de LDA.

In [33]:
```
# Pega as 10 palavras alvo de cada topico
topic_words = []
for i, topic in enumerate(lda_model_tfidf.components_):
    word_idx = topic.argsort()[::-1][:10]
    topic_words.append(" ".join([feature_names[w] for w in word_idx]))
```

In [34]:
```
# Cria o mapa de topico, indice e nome
topic_mapping = {i: name for i, name in enumerate(topic_words)}
```

In [35]:
```
def get_topic_index(tfidf: np.array):
    """
    Get topic for a lda trained model
    """

    # Calcula a probabilidade do topico de maior peso
    topic_distribution = lda_model_tfidf.transform(tfidf.reshape(1, -1))

    return np.argmax(topic_distribution)

news_2016['topic'] = news_2016.tfidf.progress_map(get_topic_index)
```

```
100%                                                  7943/7943 [00:05<00:00, 1418.41it/s]
```

In [36]:
```
# Busca o nome de cada topico baseado em seu indice
news_2016['topic_name'] = news_2016.topic.map(topic_mapping)
```

In [37]:
```
news_2016.head()
```

Out[37]:

	title	text	date	category	subcategory	link	nltk_tokens	spacy_doc
0	Fazendeira cria própria rede de banda larga e ...	"Sou apenas a mulher de um fazendeiro", diz Ch...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[sou, apen, a, mulh, de, um, fazend, diz, chri...	(", Sou, apenas, a, mulher, de, um, fazendeiro...
1	Alteração na cobrança do ICMS eleva conta de c...	A conta do celular pós-pago ou controle ficará...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[a, cont, do, celul, ou, control, fic, mais, c...	(A, conta, do, celular, pós-pago, ou, controle...
2	Ajustes sobre servidores públicos emperram nos...	A maior parte dos projetos de ajuste das conta...	2016-12-31	mercado	NaN	http://www1.folha.uol.com.br/mercado/2016/12/1...	[a, mai, part, do, projet, de, ajust, da, cont...	(A, maior, parte, dos, projetos, de, ajuste, d...

In [38]:
```
# Frequência do topico por indice
news_2016.topic.value_counts()
```

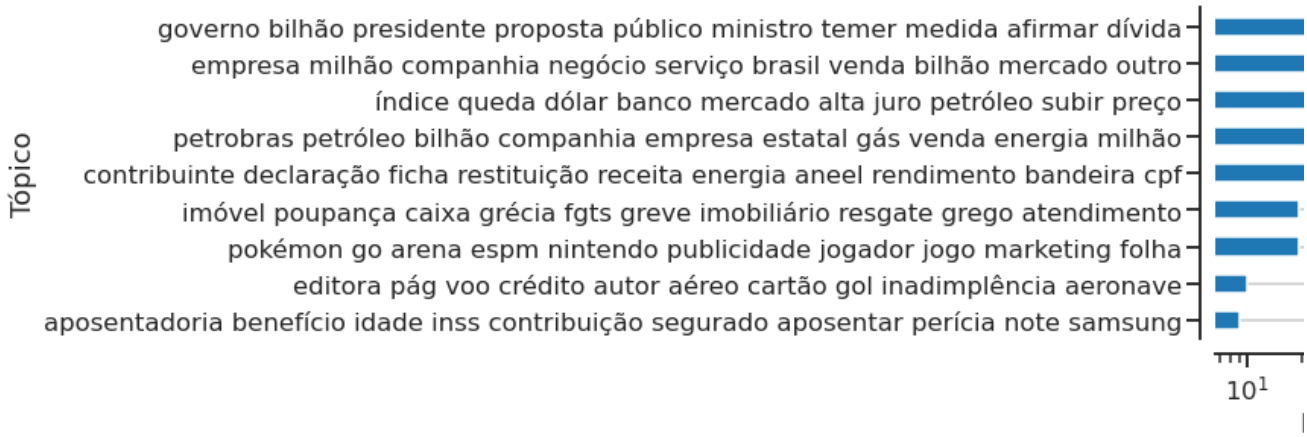
Out[38]:
```
topic
7    3136
0    2391
5    1849
3     385
4     125
6      19
1      19
2      10
8       9
Name: count, dtype: int64
```

In [39]:
```
# Frequência do topico por nome
news_2016.topic_name.value_counts()
```

Out[39]:
```
topic_name
governo bilhão presidente proposta público ministro temer medida afirmar dívida            3136
empresa milhão companhia negócio serviço brasil venda bilhão mercado outro                 2391
índice queda dólar banco mercado alta juro petróleo subir preço                            1849
petrobras petróleo bilhão companhia empresa estatal gás venda energia milhão                385
contribuinte declaração ficha restituição receita energia aneel rendimento bandeira cpf     125
pokémon go arena espm nintendo publicidade jogador jogo marketing folha                     19
imóvel poupança caixa grécia fgts greve imobiliário resgate grego atendimento               19
editora pág voo crédito autor aéreo cartão gol inadimplência aeronave                       10
aposentadoria benefício idade inss contribuição segurado aposentar perícia note samsung      9
Name: count, dtype: int64
```

In [40]:
```
with sns.axes_style("ticks"):
    sns.set_context("talk")
    ax = news_2016['topic_name'].value_counts().sort_values().plot(kind = 'barh')
    ax.yaxis.grid(True)
    ax.set_ylabel("Tópico")
    ax.set_xlabel("Número de notícias (log)")
    sns.despine(offset = 10)
    ax.set_xscale("log")
```

```
C:\Users\herik\anaconda3\lib\site-packages\seaborn\rcmod.py:400: DeprecationWarning: distutils Version classes a
  if LooseVersion(mpl.__version__) >= "3.0":
C:\Users\herik\anaconda3\lib\site-packages\setuptools\_distutils\version.py:351: DeprecationWarning: distutils V
  other = LooseVersion(other)
```

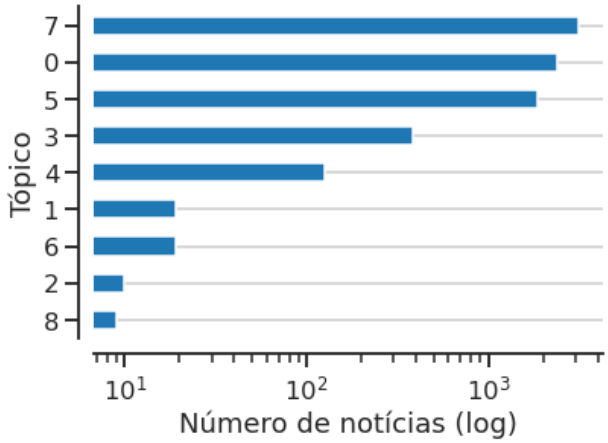


Número de documentos vs tópicos

Esse gráfico nos mostra quantos documentos foram caracterizados por cada tópico.

In [41]:
```
with sns.axes_style("ticks"):
    sns.set_context("talk")
    ax = news_2016['topic'].value_counts().sort_values().plot(kind = 'barh')
    ax.yaxis.grid(True)
    ax.set_ylabel("Tópico")
    ax.set_xlabel("Número de notícias (log)")
    sns.despine(offset = 10)
    ax.set_xscale("log")
```

```
C:\Users\herik\anaconda3\lib\site-packages\seaborn\rcmod.py:400: DeprecationWarning: distutils Version classes a
  if LooseVersion(mpl.__version__) >= "3.0":
C:\Users\herik\anaconda3\lib\site-packages\setuptools\_distutils\version.py:351: DeprecationWarning: distutils V
  other = LooseVersion(other)
```



Crie uma nuvem de palavra para cada tópico.

Use as colunas `spacy_lemma e topic` para essa tarefa.

16. Cole a figura com a nuvem de palavras para cada um dos 9 tópicos criados.

A nuvem de palavras foi gerada utilizando Wordcloud, utilizando **todas** palavras categorizadas pelo LDA.

In [42]:
```
fig, axs = plt.subplots(3, 3, figsize=(18, 18))

for idx_topico in range(9):
    # Calcula a respectiva linha e coluna do plot
    row = idx_topico // 3
    col = idx_topico % 3

    # Constroi a lista de palavras da wordcloud
    topic_news = news_2016[news_2016['topic'] == idx_topico]
    list_of_words = chain(*topic_news.spacy_lemma.values.tolist())
    string_complete = ' '.join(list_of_words)

    # Alimenta a wordcloud e joga no subplot
    wordcloud = WordCloud(width=400, height=400).generate_from_text(string_complete)
    axs[row, col].imshow(wordcloud, interpolation='bilinear')
    axs[row, col].axis('off')
    axs[row, col].set_title(f"Tópico {idx_topico+1}")

plt.tight_layout()
plt.show()
```


Tópico 1
empresa
afirmar
outro
mercado
companhia
país
Brasil
algum
pessoa
produto
venda
negócio
serviço
cliente
caso
dever
banco
Estados Unidos
Tópico 2
caixa eletrônico
banco oferecer
Internet Banking
salário mínimo
paralisação
reajuste inflação
atendimento telefone
salário benefício
greve
pagar conta
benefício abono
defesa Consumidor
equivalente salário
central atendimento
pagamento conta
reajuste salário
bancário agência
contas cliente
acordo
aplicativo caixa
Banking celular
aumento real
Tópico 4
empresa
Petrobra
bilhão
companhia
estatal
venda
Brasil
mercado
projeto
produção
operação
investimento
afirmar
país
acordo
ativo
informar
outro
Tópico 5
declaração
declarar
ficha rendimento
contribuinte
Bens direito
pagar
dependente
isento tributável
rendimento isento
ficha Bens
imóvel
receber
rendimento tributável
Imposto Renda
Receita Federal
Tópico 7
Arena Marketing
Marketing evento
folha parceria
faculdade ESPM
site tv
vivo site
tv Folha
rodada Arena
transmissão vivo
evento voltar
parceria faculdade
Brasil programa
ESPM realizar
jornalista Mariana
Mariana Barbosa
programa transmissão
publicidade propaganda
universo publicidade
mediação jornalista
Tópico 8
governo
afirmar
empresa
país
outro
Brasil
banco
acordo
dever
hoje
caso
bilhão
investimento
proposta
medida
projeto
mercado
Michel Temer
governo federal

Crie uma nuvem de entidades para cada tópico.

Use as colunas `spacy_lemma e topic` para essa tarefa.

11. Cole a figura gerada que mostra a nuvem de entidades para cada tópico obtido (no final do notebook)

Nuvem de palavras geradas apartir somente das entidades do tipo organização (filtradas anteriormente conforme solicitado).

In [43]:
```
fig, axs = plt.subplots(3, 3, figsize=(18, 18))

for idx_topico in range(9):
    # Calcula a respectiva linha e coluna do plot
    row = idx_topico // 3
    col = idx_topico % 3

    # Constroi a lista de palavras da wordcloud
    topic_news = news_2016[news_2016['topic'] == idx_topico]
    list_of_docs = topic_news.spacy_ner.apply(lambda l : [w.replace(" ", "_") for w in l])
    list_of_words = chain(*list_of_docs)
    string_complete = ' '.join(list_of_words)

    # Alimenta a wordcloud e joga no subplot
    wordcloud = WordCloud(width=400, height=400).generate_from_text(string_complete)
    axs[row, col].imshow(wordcloud, interpolation='bilinear')
    axs[row, col].axis('off')
    axs[row, col].set_title(f"Tópico {idx_topico+1}")

plt.tight_layout()
plt.show()
```

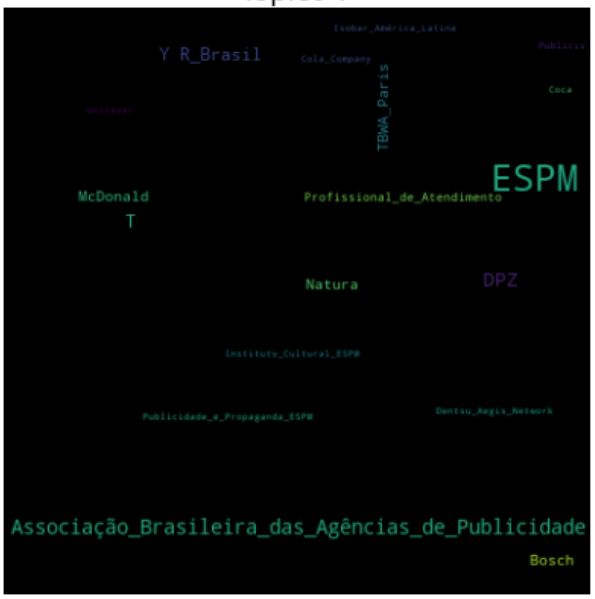
Tópico 1

Tópico 2

Tópico 4

Tópico 5

Tópico 7



Tópico 8

Perguntas Extras:

12. Quando adotamos uma estratégia frequentista para converter textos em vetores, podemos fazê-lo de diferentes maneiras. Mo IDF. Explique a principal motivação em adotar TF-IDF frente as duas outras opções.

A principal vantagem do TF-IDF é o tratamento da imporância de cada termo. O One-Hot é binário, indicando 0 ou 1 de acordo com a exist frequência. Já o TF considera a frequência de cada termo, contabilizando o numero de ocorrências de cada um.

Por ultimo o TF-IDF junta a frequência dos termos (TF) e a sua importância para o documento (IDF), dessa forma analisa o quanto um tern dando maior peso para os termos que aparecem menos e são mais distintos nos documentos.

Assim o TF-IDF consegue trazer mais valor para o documento analisado, dessa forma não é tão influenciado pelo viés de numero de palav valor e ser usado até para tarefas como analise de sentimentos e identificação de spam em emails.

17. Escreva brevemente uma descrição para cada tópico extraído. Indique se você considera o tópico extraído semanticamente c

['quanto', 'editora', 'pág', 'whatsapp', 'volkswagen', 'investigação', 'autor', 'montadora', 'pokémon', 'justiça']
Tópico 1: Aparenta se referir a investigações e noticias da investigação de montadoras, como por exemplo a vokswagen. A semântica não termos.

['de', 'empresa', 'por', 'bilhão', 'banco', 'país', 'brasil', 'milhão', 'mercado', 'em']
Tópico 2: O tópico se refere a economia e industria, como por exemplo bancos. Esse tópico já parece ter uma coesão alta entre os termos

['petrobras', 'de', 'energia', 'petróleo', 'estatal', 'gás', 'distribuidora', 'eletrobras', 'usina', 'combustível']
Tópico 3: Esse tópico se refere ao mercado de energia, gas e petroleo, termos que estão relacionados entre-si, de forma que possui uma a energia.

['uber', 'carro', 'motorista', 'veículo', 'estácio', 'kroton', 'autônomo', 'montadora', 'state', 'tesla']
Tópico 4: O tópico se refere a industria de automoveis, não necesariamente só montadoras mas veiculos em geral, realizando menção a m Uber. A coesão semântica é alta, entretanto possui alguns termos não relacionados ao tópico em geral.

['contribuinte', 'declaração', 'restituição', 'receita', 'lote', 'cpf', 'ir', 'de', 'malha', 'fino']
Tópico 5: Esse tópico se refere ao pagamento de impostos do contribuinte, principalmente sobre o período de pagamento do imposto de re

['de', 'índice', 'dólar', 'alta', 'juro', 'bolsa', 'mercado', 'on', 'banco', 'subir']
Tópico 6: Focado na macro-economia se referindo a bolsa de valores, alta do dolar e juros. Semelhante ao tópico 2, entretanto o tópico 2 t coesão entre os termos.

['petróleo', 'opep', 'barri', 'produção', 'irã', 'saudita', 'de', 'arábia', 'projeção', 'focus']
Tópico 7: O tópico 7 refere-se ao mercado internacional de petróleo, focado nos paises arabes e sua importação. O tópico possui alta coes

['oi', 'telecom', 'credor', 'anatel', 'cebrap', 'debate', 'judicial', 'invepar', 'tanure', 'pharol']
Tópico 8: Esse refere-se as telecomunicadores como por exemplo a Oi, a coesão do tópico é menor que a dos demais apesar de ainda ter

['de', 'governo', 'proposta', 'temer', 'ministro', 'bilhão', 'por', 'público', 'presidente', 'federal']
Tópico 9: O tópico se refere a política e ao governo, especificamente sobre a presidência e seus ministros. A coesão do termo também é a

18. Neste projeto, usamos TF-IDF para gerar os vetores que servem de entrada para o algoritmo de LDA. Quais seriam os passos

Como já foi realizado a limpeza dos textos durante o notebook, para gerar os vetores baseados na técnica de Doc2Vec, será necessário pr

Pressupondo que já foram removidas as stopwords, pontuações, termos indesejaveis, espaços, convertido para caixa baixa. Será necessá Doc2Vec. Os proximos passos seriam:

- Separar os dados em base de treino e teste;
- Gerar as tagas para cada documento, então seria necessario gerar um objeto com as palavras daquele documento e a tag representa
- Instanciar o objeto de Doc2Vec, nesse momento deve ser parametrizado o objeto, como por exemplo: Algoritmo a ser usado, frequênc ruido a serem consideradas, conjunto;
- Construção do vocabulario usando objeto instanciado anteriormente;
- Treinamento do modelo;
- Criação de um modelo para classificação das tags, como por exemplo Regressão Logistica.

19. Em uma versão alternativa desse projeto, optamos por utilizar o algoritmo de K-Médias para gerar os clusters (tópicos). Qual adequada como processo de vetorização? Justifique com comentários sobre dimensionalidade e relação semântica entre docum Algorithm" (https://multithreaded.stitchfix.com/blog/2016/05/27/lda2vec/#topic=38&lambda=1&term= (https://multithreaded.stitchfix.com/blog/2016/05/27/lda2vec/#topic=38&lambda=1&term=)) .

TF-IDF é uma tecnica de vetorização estatistica, de forma que será multiplicado o número de vezes que uma palavra em um documento pe resultando em um valor estatistico para cada palavra em cada documento, dessa forma o algoritmo pode não ter resultados satisfatorios co Doc2Vec é um algoritmo basado em redes neurais, compreendendo o valor semântico entre as palavras.

O algoritmo K-Means irá realizar o agrupamento e a de documentos que possuam similaridade entre si, a similaridade será calculada base para compreender a similaridade o Doc2Vec é mais adequado, visto que não irá trazer somente uma medida estatistica, mas sim o valor s

20. O algoritmo lda2vec pretende combinar o poder do word2vec com a interpretabilidade do algoritmo LDA. Em qual cenário o a novo algoritmo?

O algoritmo lda2vec une o poder do word2vec e LDA, dessa forma é possível realizar uma analista mista, analisando globalmente os docu analise individual por palavra e geral dos documentos, realizando identificação de contexto com maior precisão.

O autor comenta que o principal beneficio do algoritmo lda2vec é visto quando quer construir modelos para humanos e não para maquinas tópicos centrais para cada sentença. Ainda assim o autor não recomenda o uso do lda2vec, mantendo a recomendação do LDA quando se para maquinas.

Também é comentado que o algoritmo não possui uma implementação tão fácil quanto o word2vec ou LDA, além de possuir um custo com recomendado o uso de GPUs para a sua execução.