Processamento de linguagem natural com Python [25E2_2]

Aluna: Rachel Reuters

Implementar técnicas de lematização

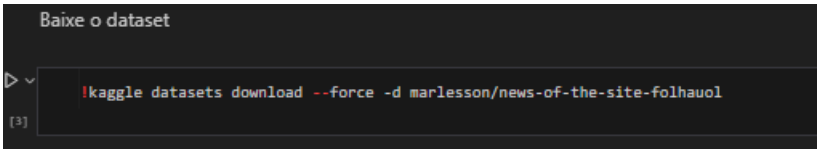
1) Qual o endereço do seu notebook (colab) executado? Use o botão de compartilhamento do colab para obter uma url.

   ➔https://github.com/rachelreuters/posGraduacaoIA/blob/main/NLP/Projeto_de_Disciplina_de_Text_Mining.ipynb

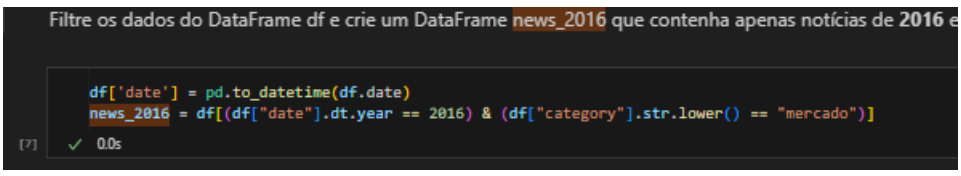
2) Em qual célula está o código que realiza o download dos pacotes necessários para tokenização e stemming usando nltk?

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

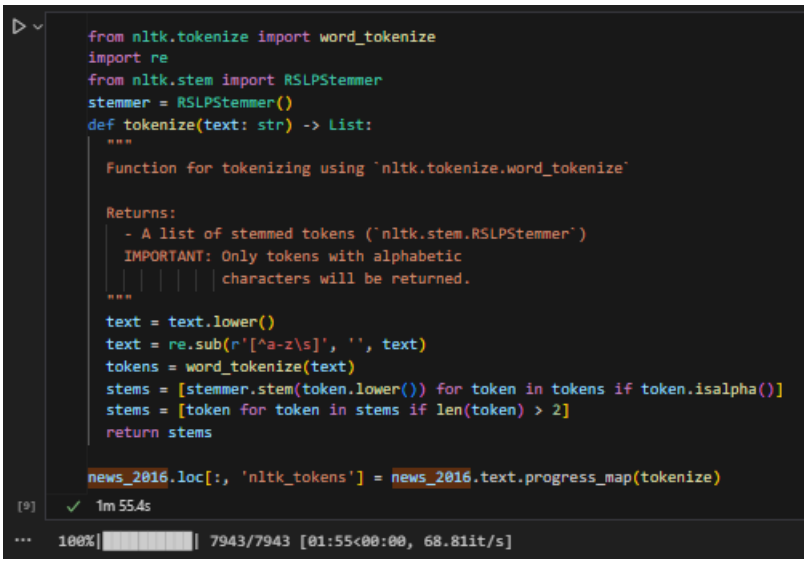
4) Em qual célula está o download dos dados diretamente do kaggle?



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



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

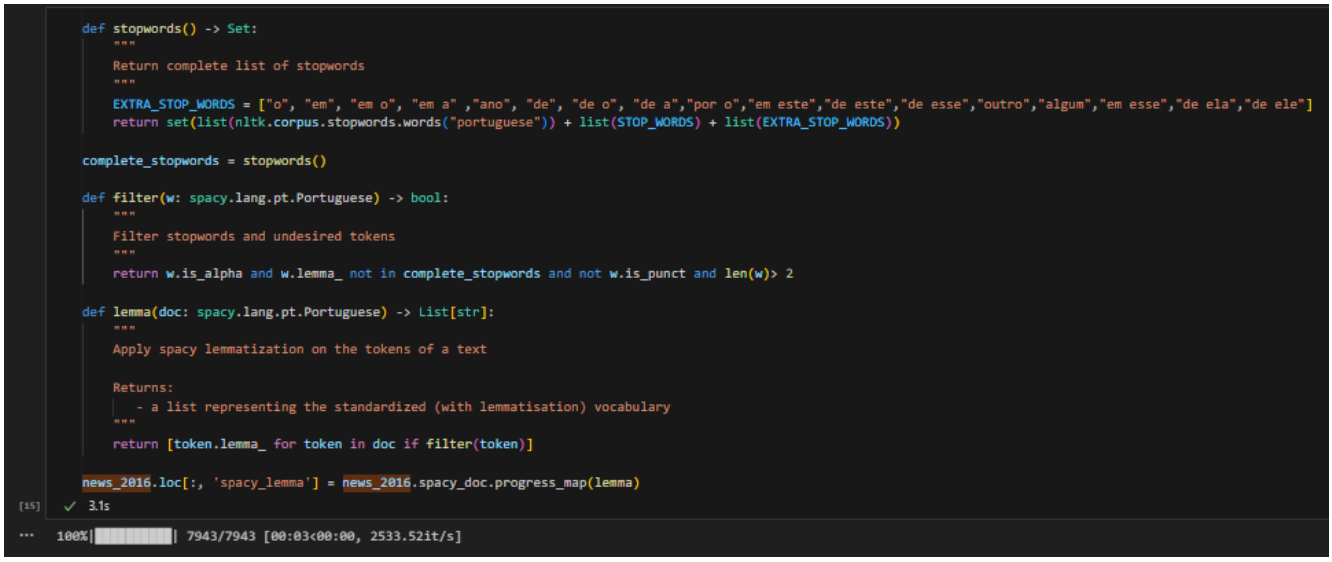


```
from nltk.tokenize import word_tokenize
import re
from nltk.stem import RSLPStemmer
stemmer = RSLPStemmer()
def tokenize(text: str) -> List:
    """
    Function for tokenizing using `nltk.tokenize.word_tokenize`

    Returns:
        - A list of stemmed tokens (`nltk.stem.RSLPStemmer`)
        IMPORTANT: Only tokens with alphabetic
                   characters will be returned.
    """
    text = text.lower()
    text = re.sub(r'[^a-z\s]', '', text)
    tokens = word_tokenize(text)
    stems = [stemmer.stem(token.lower()) for token in tokens if token.isalpha()]
    stems = [token for token in stems if len(token) > 2]
    return stems

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

[9] ✓ 1m 55.4s

... 100%|██████████| 7943/7943 [01:55<00:00, 68.81it/s]

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



```
def stopwords() -> Set:
    """
    Return complete list of stopwords
    """
    EXTRA_STOP_WORDS = ["o", "em", "em o", "em a" ,"ano", "de", "de o", "de a","por o","em este","de este","de esse","outro","algum","em esse","de ela","de ele"]
    return set(list(nltk.corpus.stopwords.words("portuguese")) + list(STOP_WORDS) + list(EXTRA_STOP_WORDS))

complete_stopwords = stopwords()

def filter(w: spacy.lang.pt.Portuguese) -> bool:
    """
    Filter stopwords and undesired tokens
    """
    return w.is_alpha and w.lemma_ not in complete_stopwords and not w.is_punct and len(w)> 2

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

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

[15] ✓ 3.1s

... 100%|██████████| 7943/7943 [00:03<00:00, 2533.52it/s]

8) Baseado nos resultados qual a diferença entre stemming e lematização, qual a diferença entre os dois procedimentos? Escolha quatro palavras para exemplificar.

a) O **stemming** é um processo mais heurístico e mais simples que remove sufixos (e às vezes prefixos) das palavras para obter o seu radical (ou stem). Ele não se preocupa se o radical resultante é uma palavra válida no dicionário.
Já a **lematização** é um processo mais sofisticado que visa reduzir uma palavra à sua forma canônica ou de dicionário. Este processo leva em consideração a morfologia da palavra e o seu contexto na frase.

b) Alguns exemplos:

Stemming	Lematização	Palavra original	Onde encontrar
mor	morador	moradores	1º documento
veloc	velocidade	velocidades	1º documento
condi	condição	condições	1º documento
telefn	telefônico	telefônica	1º documento

Construir um modelo de reconhecimento de entidades (NER) usando Spacy

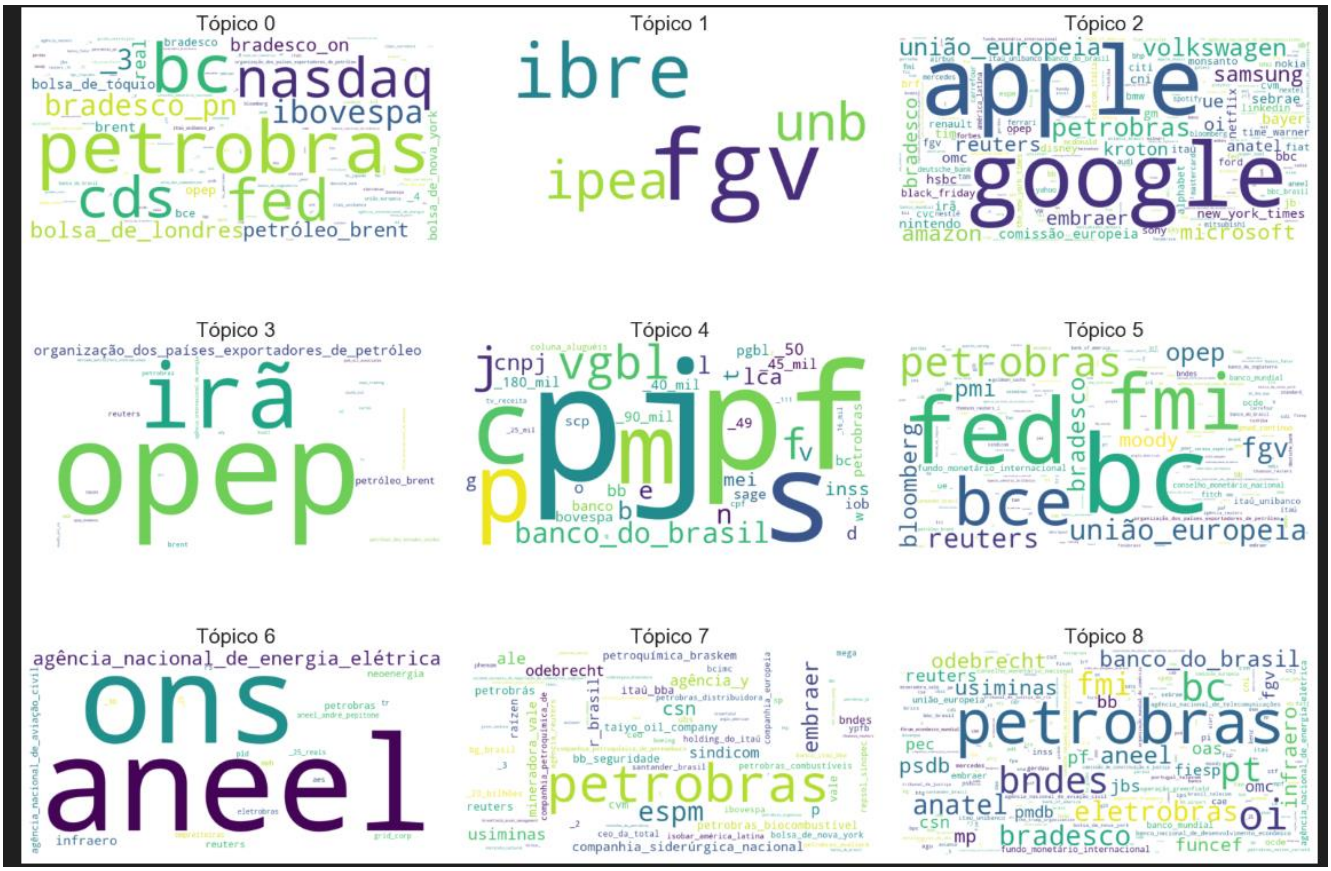
9) Em qual célula o modelo pt_core_news_lg está sendo carregado? Todos os textos do dataframe precisam ser analisados usando os modelos carregados. Em qual célula isso foi feito?

```
import spacy

nlp = spacy.load("pt_core_news_lg")

news_2016["spacy_doc"] = list(nlp.pipe(news_2016.text.str.lower(), batch_size=50))
[12]  ✓ 6m 32.0s
```

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

```
def NER(doc: spacy.tokens.Doc):
    """
    Return the list of organizations for a SPACY document
    """
    return [ent.text for ent in doc.ents if ent.label_ == "ORG"]
news_2016.loc[:, 'spacy_ner'] = news_2016.spacy_doc.progress_map(NER)
[18]  ✓ 0.3s
...  100%|██████████| 7943/7943 [00:00<00:00, 27058.14it/s]
```

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



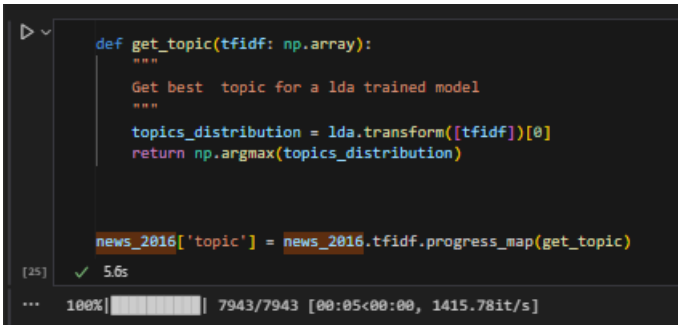
Criar modelos utilizando vetorização de textos baseado em Bag of Words

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

➔ O TF-IDF põe mais peso às palavras que são importantes dentro de 1 documento, mas não muito frequentes nos outros, ele também reduz a influência de palavras comuns, aumentando o impacto de termos específicos . Esse método também reduz o peso dos termos que são muito comuns em toda a coleção.

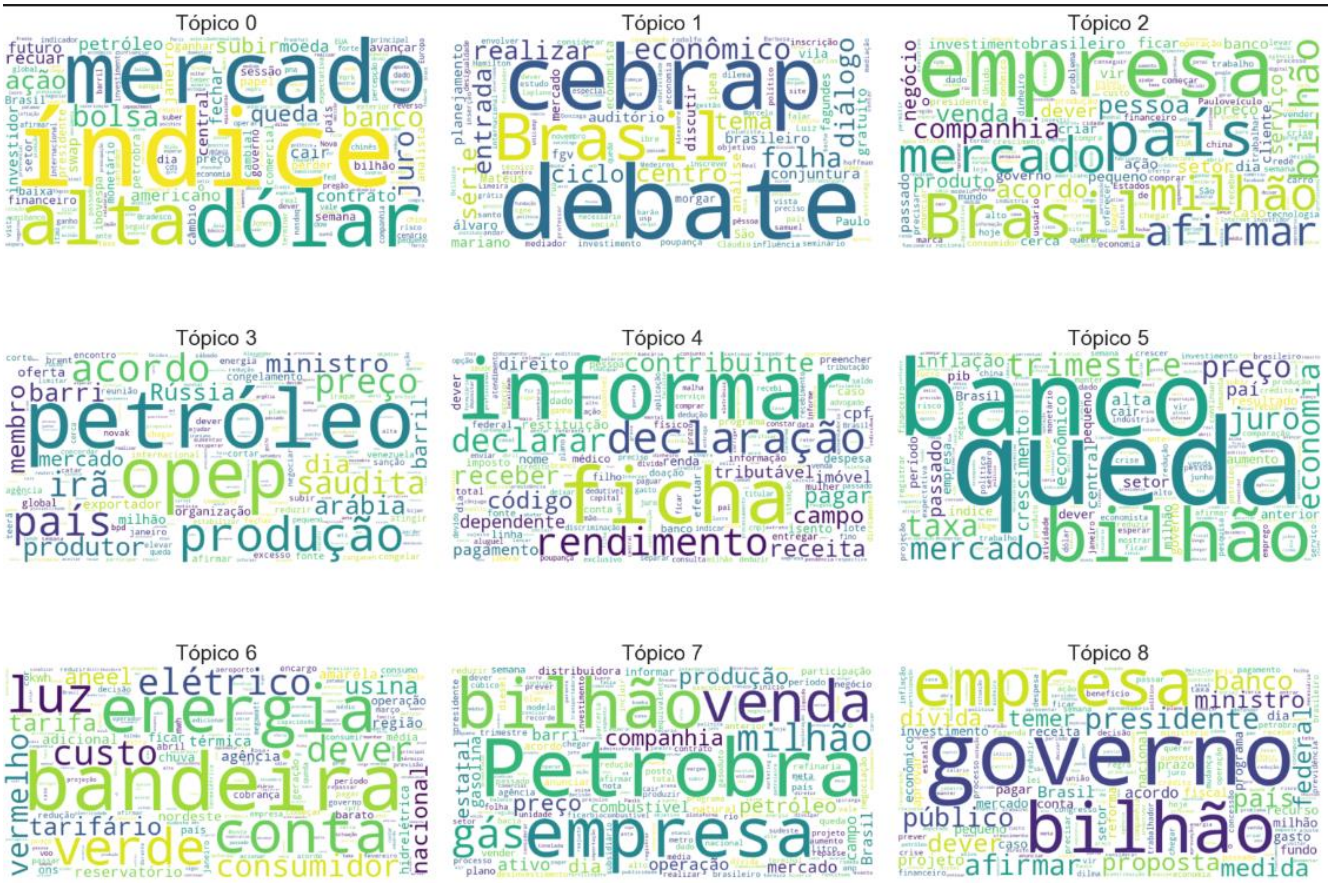
13) Indique a célula onde está a função que cria o vetor de TF-IDF para cada texto.

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



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

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



17) Escreva brevemente uma descrição para cada tópico extraído. Indique se você considera o tópico extraído semanticamente consistente ou não.

Tópico 0: Setor financeiro, talvez mercado de ações e câmbio

Tópico 1: Eventos Acadêmicos e debates (mais voltados para o lado Político-Sociais)

Tópico 2: Atividade Empresarial e Comércio Nacional

Tópico 3: Setor de petróleo e geopolítica energética

Tópico 4: Declaração de Imposto de renda, como declarar, dicas e afins.

Tópico 5: Política monetária, inflação

Tópico 6: Setor elétrico, tarifa, conta de luz e transporte aéreo

Tópico 7: Setor de óleo e gás do mercado brasileiro, mais focado na Petrobras que é a principal estatal do país nesse ramo.

Tópico 8: Cenário político do Brasil, 3 poderes, decisões políticas.

	0	1	2	3	4	5	6	7	8
0	dólar	cebrap	empresa	petróleo	declaração	queda	energia	petrobras	governo
1	índice	debate	brasil	opep	contribuinte	trimestre	aéreo	petróleo	bilhão
2	bolsa	azeredo	milhão	irã	ficha	banco	voo	petrobra	presidente
3	alta	auditório	país	barri	restituição	taxa	aneel	gás	proposta
4	mercado	diálogo	negócio	produção	rendimento	inflação	usina	estatal	ministro
5	ação	conjuntura	mercado	saudita	receita	economia	elétrico	companhia	público
6	juro	mariano	companhia	btg	cpf	juro	passageiro	empresa	temer
7	moeda	fgv	venda	arábia	declarar	crescimento	distribuidora	produção	dívida
8	petróleo	gratuito	produto	barril	dependente	preço	aviação	venda	medida
9	subir	inscrição	serviço	produtor	informar	bilhão	gol	combustível	federal

Criar modelos baseados em Word Embedding

18) Neste projeto, usamos TF-IDF para gerar os vetores que servem de entrada para o algoritmo de LDA. Quais seriam os passos para gerar vetores baseados na técnica de Doc2Vec?

1. Coletar um conjunto de documentos de texto
2. Tokenizar os textos
3. Remover stopwords
4. Criação e Treinamento do Modelo Doc2Vec
5. Transformar os documentos em vetores
6. Aplicação dos Vetores: Os vetores gerados podem ser usados para clusterização com K-Means, DBSCAN, classificação de textos (SVM, Random Forest) ou treinamento de modelos de deep learning.

19) Em uma versão alternativa desse projeto, optamos por utilizar o algoritmo de K-Médias para gerar os clusters (tópicos). Qual das abordagens (TF-IDF ou Doc2Vec) seria mais adequada como processo de vetorização? Justifique com comentários sobre dimensionalidade e relação semântica entre documentos.

➔ O processo de vetorização mais adequado no caso de utilizar um algoritmo de clusterização tipo o Kmeans seria o Doc2Vec pois reduz a dimensionalidade e torna a abordagem mais escalável. Além disso essa abordagem leva em consideração a relação semântica entre as palavras pois captura o contexto em que a palavra está inserida.

➔Ao final do notebook, eu apliquei o algoritmo de Kmeans nos vetores gerados e também com redução de dimensionalidade para 2 dimensões. Os resultados foram bem próximos ao TF-IDF.

20) Leia o artigo "Introducing our Hybrid lda2vec Algorithm" (https://multithreaded.stitchfix.com/blog/2016/05/27/lda2vec/#topic=38&lambda=1&term=) .
O algoritmo lda2vec pretende combinar o poder do word2vec com a interpretabilidade do algoritmo LDA. Em qual cenário o autor sugere que há benefícios para utilização deste novo algoritmo?

➔ Essa abordagem requer um esforço computacional bem grande, portanto mais indicada quando a quantidade de dados é muito elevada. Cenários onde os tópicos precisam ter relações mais profundas com documentos ou usuários. Uma utilização indicada para esse modelo seria personalizar um sistema de recomendação de artigos baseado nos interesses do usuário ou análise de tendencia de um blog ao longo dos anos.