ATIVIDADE FINAL PARA ENTREGA EM GRUPO

3) Reproduzir em código Python uma solução equivalente ao apresentado pelos pesquisadores do artigo, inclusive usando as mesmas técnicas, a fim de gerar uma solução para o problema em questão.
4) Comentar o código, em especial os recursos de NLP usados para o processamento do texto e comparar os achados com eventuais resultados apresentados pelos autores.

Universidade:	Nove de Julho
Programa:	PPGI - Programa de Pós Graduação em Informática
Disciplina:	PROCESSAMENTO DE LINGUAGEM NATURAL - CONCEITOS E APLICAÇÕES
Professor:	Prof. Dr. Cleber Gustavo Dias
Alunos:	André L.M.F.Santos (**RA:622150026**); Maria Fátima B.Marques (**RA:622250106**); e Reinaldo R. Oliveira (**RA:622250107**)

In [1]:
```
import pandas as pd
import numpy as np
import os
import nltk
nltk.download('vader_lexicon')
import re
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\99836932\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

In [2]:
```
# aponta o caminho onde estão os dados
os.chdir('C:/Users/99836932/Downloads/PPGI_Hotel/TopicModel')
# ler os dados
reviews_df = pd.read_csv('Hotel_Reviews.csv')
print(len(reviews_df))
print('Máximo Average Score =', max(reviews_df.Average_Score))
reviews_df.head()
```

```
515738
Máximo Average Score = 9.8
```

Out[2]:

	Hotel_Address	Additional_Number_of_Scoring	Review_Date	Average_Score	Hotel_Name	Reviewer_Nationality	Negative_Review	Review_Total_Negat
0	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	8/3/2017	7.7	Hotel Arena	Russia	I am so angry that i made this post available...	
1	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	8/3/2017	7.7	Hotel Arena	Ireland	No Negative	
2	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/31/2017	7.7	Hotel Arena	Australia	Rooms are nice but for elderly a bit difficul...	
3	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/31/2017	7.7	Hotel Arena	United Kingdom	My room was dirty and I was afraid to walk ba...	
4	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/24/2017	7.7	Hotel Arena	New Zealand	You When I booked with your company on line y...	

In [3]:

```
# Top 10 hotéis
top10 = reviews_df.groupby(['Hotel_Name'])['Average_Score'].mean().reset_index()
top10 = top10.sort_values(by = ['Average_Score'], ascending = False)
top10.head(10)
```

Out[3]:

	Hotel_Name	Average_Score
1202	Ritz Paris	9.8
3	41	9.6
472	H tel de La Tamise Esprit de France	9.6
499	Haymarket Hotel	9.6
771	Hotel The Serras	9.6
481	H10 Casa Mimosa 4 Sup	9.6
598	Hotel Casa Camper	9.6
1015	Milestone Hotel Kensington	9.5
1303	Taj 51 Buckingham Gate Suites and Residences	9.5
750	Hotel Sacher Wien	9.5

In [4]:

```
# Cria uma coluna juntando os Reviews negativos e positivos
reviews_df["review"] = reviews_df["Negative_Review"] + reviews_df["Positive_Review"]
# cria uma coluna de rótulos de reviews ruins (1), zero caso contrário
reviews_df["is_bad_review"] = reviews_df["Reviewer_Score"].apply(lambda x: 1 if x < 5 else 0)
# seleciona apenas as colunas de interesse
reviews_df = reviews_df[["review", "is_bad_review", "Hotel_Name"]]
# Seleciona apenas se não for bad_review
reviews_df = reviews_df[reviews_df.is_bad_review == 0] # NPS Muito bom ou Excelente (acima de 5 pontos)
print(len(reviews_df))
reviews_df.head()
```

```
493457
```

Out[4]:

	review	is_bad_review	Hotel_Name
1	No Negative No real complaints the hotel was g...	0	Hotel Arena
2	Rooms are nice but for elderly a bit difficul...	0	Hotel Arena
4	You When I booked with your company on line y...	0	Hotel Arena
5	Backyard of the hotel is total mess shouldn t...	0	Hotel Arena
7	Apart from the price for the brekfast Everyth...	0	Hotel Arena

In [5]:
```
# Concatena os reviews por hotel
reviews_df = reviews_df.groupby('Hotel_Name').agg({'review': lambda x: ''.join(x)}).reset_index()
reviews_df
```

Out[5]:

	Hotel_Name	review
0	11 Cadogan Gardens	Thought the prise of drinks at the bar a litt...
1	1K Hotel	On a main street so can hear the traffic book...
2	25hours Hotel beim MuseumsQuartier	Breakfast not included and buffet really expe...
3	41	There wasn t a thing that we didn t like Its...
4	45 Park Lane Dorchester Collection	More kinds of fruit juice will make the mini ...
...	...	...
1487	citizenM London Bankside	This was our third stay at this hotel and it ...
1488	citizenM London Shoreditch	Lifts need reprogramming exasperating journey...
1489	citizenM Tower of London	Rooms are small but well designed The use of ...
1490	every hotel Piccadilly	The hotel overall requires an update furnitur...
1491	pentahotel Vienna	Slightly high prices at the bar and odd smell...

1492 rows × 2 columns

In [6]:
```
# Confere o total de Hotéis na base
len(reviews_df.Hotel_Name.unique())
```

Out[6]:
```
1492
```

In [7]:
```
# Cria uma dataframe para verificar se há algum hotel repetido
dh = pd.DataFrame(reviews_df.groupby(['Hotel_Name']).size()).reset_index()
dh.columns = ['Hotel','Total']
dh.sort_values(by = ['Total'], ascending = False)
```

Out[7]:

	Hotel	Total
0	11 Cadogan Gardens	1
992	Mercure Paris Gare Montparnasse	1
1001	Mercure Paris Pigalle Sacre Coeur	1
1000	Mercure Paris Opera Louvre	1
999	Mercure Paris Opera Grands Boulevards	1
...	...	...
494	Ham Yard Hotel	1
493	Hallmark Hotel London Chigwell Prince Regent	1
492	HCC St Moritz	1
491	HCC Regente	1
1491	pentahotel Vienna	1

1492 rows × 2 columns

In [8]:
```
# Amostra
#reviews_df = reviews_df.sample(frac = 0.01, replace = False, random_state=42)
```

Limpeza dos dados

In [9]:
```
# remove 'No Negative' or 'No Positive' from text
reviews_df["review"] = reviews_df["review"].apply(lambda x: x.replace("No Negative", "").replace("No Positive", ""))
```

In [10]:
```
# Preprocessamento do corpus (reviews)
#import nltk
#import numpy as np

# Remove stop words em inglês
wpt = nltk.WordPunctTokenizer()
stop_words = nltk.corpus.stopwords.words('english')
```

```
# Função para normalizar o corpus
def normalize_document(doc):
    # letras minúsculas e remover caracteres especiais\espaço em branco
    doc = re.sub(r'[^a-zA-Z\s]', '', doc, re.I|re.A)
    doc = doc.lower()
    doc = doc.strip()
    # tokenizar documento
    tokens = wpt.tokenize(doc)
    # filtrar stopwords do documento
    filtered_tokens = [token for token in tokens if token not in stop_words]
    # recriar documento a partir de tokens filtrados
    doc = ' '.join(filtered_tokens)
    return doc

normalize_corpus = np.vectorize(normalize_document)
```

In [11]:
```
# Normalizar o documento
#import re
reviews_df["review_clean"] = normalize_corpus(reviews_df["review"])
print(len(reviews_df))
reviews_df.head()
```

```
1492
```

Out[11]:

	Hotel_Name	review	review_clean
0	11 Cadogan Gardens	Thought the prise of drinks at the bar a litt...	thought prise drinks bar little excessive part...
1	1K Hotel	On a main street so can hear the traffic book...	main street hear traffic booked double room be...
2	25hours Hotel beim MuseumsQuartier	Breakfast not included and buffet really expe...	breakfast included buffet really expensive coo...
3	41	There wasn t a thing that we didn t like Its...	thing like central proximity close services re...
4	45 Park Lane Dorchester Collection	More kinds of fruit juice will make the mini ...	kinds fruit juice make mini bar better everyth...

TF-IDF Features

In [12]:
```
import sklearn
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```
# Converter os textos em vetores TF-IDF
vectorizer = TfidfVectorizer()
vectors = vectorizer.fit_transform(reviews_df["review_clean"])

# Calcule a similaridade do cosseno entre os vetores
similarity = cosine_similarity(vectors)
similarity
```

Out[12]:

```
array([[1.        , 0.64662591, 0.68631022, ..., 0.64874971, 0.76842526,
        0.6131129 ],
       [0.64662591, 1.        , 0.685016  , ..., 0.62981457, 0.69350359,
        0.63295608],
       [0.68631022, 0.685016  , 1.        , ..., 0.70519119, 0.72103617,
        0.73232129],
       ...,
       [0.64874971, 0.62981457, 0.70519119, ..., 1.        , 0.701012  ,
        0.60189158],
       [0.76842526, 0.69350359, 0.72103617, ..., 0.701012  , 1.        ,
        0.63689053],
       [0.6131129 , 0.63295608, 0.73232129, ..., 0.60189158, 0.63689053,
        1.        ]])
```

In [13]:

```
# Adcionar nome dos hotéis nas colunas
sim = pd.DataFrame(similarity)
new_labels = reviews_df.Hotel_Name.to_list()
sim = sim.set_axis(new_labels, axis=1)
sim.index = new_labels
sim
```

Out[13]:

	11 Cadogan Gardens	1K Hotel	25hours Hotel beim MuseumsQuartier	41	45 Park Lane Dorchester Collection	88 Studios	9Hotel Republique	A La Villa Madame	ABaC Restaurant Hotel Barcelona GL Monumento	AC Hotel Barcelona Forum a Marriott Lifestyle Hotel	...	XO Hotel	B
11 Cadogan Gardens	1.000000	0.646626	0.686310	0.617111	0.378445	0.588584	0.700980	0.510188	0.395453	0.584496	...	0.439498	
1K Hotel	0.646626	1.000000	0.685016	0.465892	0.294538	0.612146	0.735616	0.472871	0.374625	0.610718	...	0.497075	
25hours Hotel beim MuseumsQuartier	0.686310	0.685016	1.000000	0.565722	0.370641	0.564934	0.716394	0.515286	0.436234	0.665228	...	0.460019	
41	0.617111	0.465892	0.565722	1.000000	0.400499	0.394822	0.511272	0.400319	0.317764	0.450526	...	0.327376	
45 Park Lane Dorchester Collection	0.378445	0.294538	0.370641	0.400499	1.000000	0.266916	0.320281	0.249186	0.229111	0.281446	...	0.222641	
...	...	...	...	...	...	...	...	...	...	...	...	...	
citizenM London Bankside	0.691547	0.679213	0.748631	0.543041	0.372153	0.637740	0.731287	0.475916	0.379353	0.621356	...	0.448320	
citizenM London Shoreditch	0.634868	0.636452	0.688079	0.479987	0.338138	0.604948	0.684939	0.446331	0.354006	0.579537	...	0.421286	
citizenM Tower of London	0.648750	0.629815	0.705191	0.501687	0.356110	0.608904	0.671456	0.444310	0.359931	0.600862	...	0.411607	
every hotel Piccadilly	0.768425	0.693504	0.721036	0.574391	0.375103	0.645464	0.731836	0.543035	0.391844	0.636491	...	0.489955	
pentahotel Vienna	0.613113	0.632956	0.732321	0.509929	0.315226	0.546391	0.667883	0.443183	0.362293	0.616382	...	0.441119	

1492 rows × 1492 columns

In [14]:
```
# Analisa quais os hotéis com maior similaridade ao Hotel Pulizer
HP = sim[['Hotel Pulitzer']]
HP = HP.sort_values(by = ['Hotel Pulitzer'], ascending = False)
```

```
HP = HP.iloc[1: , :]
HP.head(10)
```

Out[14]:

	Hotel Pulitzer
Royal Passeig de Gracia	0.906183
Hotel Espa a Ramblas	0.893724
Avenida Palace	0.893451
H10 Cubik 4 Sup	0.888973
H10 Urquinaona Plaza	0.886879
H10 Metropolitan 4 Sup	0.884674
Negresco Princess 4 Sup	0.881367
Catalonia Plaza Catalunya	0.874998
Condes de Barcelona	0.870546
Hotel Barcelona Catedral	0.869294

Considerações finais

Os 3 primeiros hotéis do nosso modelo foram os mesmos do artigo; porém, a ordem dos outros ficou diferente. Primeiro, isso aconteceu, pois atribuímos ao modelo uma nota de corte, com base no Net Promoter Score (NPS). Assim, só foram considerados as notas com NPS Muito bom ou Excelente (acima de 5 pontos).

Outra diferença pode ser em relação ao tratamento e limpeza dos dados, além da biblioteca que os autores usaram.

Referências

- Sentiment analysis with hotel reviews
- Github — TopicModel Script
- Net Promoter Score