**Objective:**

To conduct a sentiment analysis of Park Lane Hotel reviews on TripAdvisor as a way of identifying the amenities that are most often cited in negative reviews.

The first step of the process is the scrape the reviews.

**About Park Lane Hotel:**

The Park Lane Hotel is a luxury 46-story hotel with panoramic views of Central Park and the New York City skyline. Guest Rooms and Guest Suites offer picture windows and stunning views along with luxury bath products, fine linens and other amenities.

Guests at our Central Park Hotel enjoy classical ambience and service coupled with new-world comfort and convenience for an unparalleled experience in the world's most extraordinary city.

**About TripAdvisor:**

TripAdvisor is the world's largest travel platform with more than 830 million reviews and opinions of 8.6 million accommodations, restaurants, experiences, airlines, and cruises. TripAdvisor is available in 49 markets and 28 languages.

**Process:**

The Park Lane Hotel was chosen at random from 802 New York City hotels listed on TripAdvisor.

The first step in scraping the data was to isolate the review copy from the various elements in each hotel listing. This was done with the following Python script:

import re

import requests

from bs4 import BeautifulSoup

import time

import os

import pandas as pd

data = []

my\_headers = { 'User-Agent': 'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36'}

page = 100

for k in range(0, page+5, 5):

if k == 0 :

url = 'https://www.tripadvisor.com/Hotel\_Review-g60763-d93579-Reviews-Park\_Lane\_Hotel-New\_York\_City\_New\_York.html#REVIEWS'

else:

url = 'https://www.tripadvisor.com/Hotel\_Review-g60763-d93579-Reviews-or'+str(k)+'-Park\_Lane\_Hotel-New\_York\_City\_New\_York.html#REVIEWS'

src = False

for i in range(1,6):

try:

response = requests.get(url, headers = my\_headers)

src = response.content

break

except:

print('failed attempt #', i)

time.sleep(2)

if not src:

print('Could not get page', url)

else:

print('Successfully get the page', url)

soup = BeautifulSoup(src.decode('ascii', 'ignore'),'lxml')

divs = soup.findAll('div', {'class':re.compile('hotels-review-list-parts-SingleReview\_\_mainCol--2XgHm')})

for div in divs:

span = div.find('span', {'class':re.compile('ui\_bubble\_rating')})

reviews = div.find('div', {'class':re.compile('cPQsENeY')})

if span:

rating = span.attrs['class'][1]

if reviews:

review = reviews.find('span')

review\_text = review.text.strip()

data.append([rating, review\_text])

with open('review.txt', 'w', encoding='utf-8') as f:

for text in data:

f.write(text[0] + '\t' + text[1] + '\n')

This resulted in 105 reviews from 20 pages of comments.

An additional scraping was conducted using nltk to clean (remove punctuation, convert to lower case, and remove stop words) and lemmatize the comment copy. This Python script was written for that purpose:

import re

import string

import csv

from string import punctuation

import pandas as pd

import nltk

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

output\_data = pd.read\_csv('review.txt', delimiter='\t', header=None)

output\_data.columns = ["rating","review"]

for i in range(len(output\_data)):

output\_data.iloc[i]['rating'] = output\_data.iloc[i]['rating'][7:]

## remove punctuation

for i in range(len(output\_data)):

output\_data.iloc[i]['review'] = ''.join(text for text in output\_data.iloc[i]['review'] if text not in punctuation)

## convert text to lowercase

for i in range(len(output\_data)):

output\_data.iloc[i]['review'] = ' '.join([text.lower() for text in nltk.word\_tokenize(output\_data.iloc[i]['review'])])

## lemmatize

stopword = stopwords.words('english')

word\_lemmatizer = WordNetLemmatizer()

for i in range(len(output\_data)):

output\_data.iloc[i]['review'] = [word\_lemmatizer.lemmatize(w) for w in nltk.word\_tokenize(output\_data.iloc[i]['review'])]

## stop words removal

for i in range(len(output\_data)):

output\_data.iloc[i]['review'] = [word for word in output\_data.iloc[i]['review'] if word not in stopword]

print(output\_data)

word\_count = {}

for i in range(len(output\_data)):

for word in output\_data.iloc[i]['review']:

if word in word\_count:

word\_count[word] += 1

else:

word\_count[word] = 1

#Print in sorted order

for w in sorted(word\_count, key=word\_count.get, reverse=True):

print (w, word\_count[w])

with open('word\_count.csv','w') as f:

fieldnames = ['word', 'frequency']

word = csv.DictWriter(f,fieldnames=fieldnames)

word.writeheader()

for w in sorted(word\_count, key=word\_count.get, reverse=True):

word.writerow({'word': w, 'frequency': word\_count[w]})

Here is a small sample of the comments that we were able to scrape:

bubble\_40 This is a traditional older hotel located off of Central Park. The views are amazing and the hotel bar is nice (but no views). Rooms are large for New York standards, bed is comfortable but HVAC did not work well in my room. I like the great location around everything. Try to get a room with a view otherwise you will look into buildings. Bathroom is ok but small like other nyc hotels. Elevators are quick and efficient. Bathrooms on second floor by bar and restaurant are convenient but mens room door was broke so had to use Unisex bathroom which was ok. Would probably not stay here again but price is slightly lower than JW and Ritz next door.

bubble\_50 We had yet another wonderful visit at this hotel this time with a full park view. Thank you for the upgrade! We love staying here when ever we come to NYC it is a perfect location for us. Near subways and the Park everything we want to do is within walking distance. Want to go to a play we walk the museums we walk. Always clean and with helpful people on staff we always enjoy our stay here.

bubble\_20 Booked this hotel with BA, thought wed got a good deal , as photos of hotel looked great ! However, dont be deceived , its just not what youd expect from a supposedly 4 star hotel. All the carpets are filthy , and rooms are tired. Hotel reception staff were really rude when checking in. Got really offish when asked to explain the resort fee which is a total rip off. We stayed for 3 nights $35 per night, then another couple of hundred dollars on top as a retainer which is immediately taken off your credit card . But you get some of it back !! 10days later . The only good thing I can say is that its in a great location . Im sure there are much better hotels in NY, which are better value for money -without the added resort fee .

From there we identified the most frequent words in the reviews. Here are the top 10 words and their frequency:

|  |  |
| --- | --- |
| room | 139 |
| hotel | 131 |
| park | 120 |
| view | 63 |
| location | 61 |
| staff | 55 |
| great | 50 |
| lane | 50 |
| central | 49 |
| stay | 46 |

Next Steps:  
Using the most frequently cited words in the reviews that we scraped, we will build a classifier to determine which are negative and which are positive. Then we will be able to use the results of that script to analyze the most frequently cited reasons for a negative review.