Analysis of TripAdvisor Data with the help of Selenium and Python

A Practical Tutorial on Scraping and Analyzing TripAdvisor Data using Selenium and Python

Photo from https://scrapfly.io/blog/how-to-scrape-tripadvisor

Web scraping is a powerful method for extracting data from websites. TripAdvisor, with its extensive traveler reviews and ratings, offers valuable insights into customer preferences. However, manual extraction and analysis from TripAdvisor can be time-consuming.

This tutorial focuses on using automation and programming with Python and Selenium, a web scraping tool, to efficiently scrape and analyze TripAdvisor data. It also covers the use of Python libraries like Pandas and Matplotlib to clean, structure, and visualize the scraped data. Additionally, the tutorial explains how to perform sentiment analysis on the collected reviews, providing further insights into customer opinions and preferences.

• • •

Installing and setting up Selenium

To get started, you need to install the Selenium library. Open your command prompt or terminal and run the following command:

```
pip install selenium
```

Selenium requires a WebDriver to interact with web browsers. The WebDriver acts as a bridge between your Python code and the browser. The most commonly used WebDriver is for Chrome, called ChromeDriver. If you don't have the driver installed on your computer, you can download it from here.

Then, you can use the following command in your python script so that Selenium can locate it:

```
driver = webdriver.Chrome()
```

TripAdvisor dataset – What do we need to extract and how??

In this tutorial, you will learn how to scrape data for all the reviews of 'Coffee & Tea' and 'Bars & Pubs' establishments in Thessaloniki from the TripAdvisor website. For each review, you will extract the following data:

1. Business Reviewed: The name of the establishment that was reviewed.
2. Reviewer Username: The username of the reviewer who left the review.
3. Review Date: The date when the review was posted.
4. Visit Date: The date of the reviewer's visit to the establishment.
5. Review Title: The title or heading of the review.
6. Review Text: The main content of the review.
7. Review Rating: The rating given by the reviewer to the establishment.

To extract the data, we'll be using Python and the Selenium library. Make sure you have Python and Selenium installed and the appropriate WebDriver for your browser.

You will use this url to extract the information needed. To verify that the setup is working correctly, you can execute the following script:

```
from selenium import webdriver

url = 'https://www.tripadvisor.com/Restaurants-g189473-Thessaloniki_Thessaloniki_Region_Central_Macedonia.html'
driver = webdriver.Chrome()

# Open the website
driver.get(url)

# Close the browser
driver.quit()
```

The scraping process involves two primary while loops. Initially, you will iterate through each page of the search results by clicking on the pagination buttons. This allows you to access additional pages and ensure you retrieve reviews from all pages. For each page, you will open each review in a new tab and extract the desired information. The information

from the reviews will be stored in a dictionary. After iterating through all the businesses and collecting the data, you will store the dictionary into a CSV file.

• • •

TripAdvisor dataset – Web scraping process

In more detail, in order to scrape the aforementioned data, the actions that should take place are the following:

1. Click the **"Show more"** element in the url page.
2. Find and click checkboxes for **"Coffee & Tea"**, **"Bars & Pubs"** and unclick **"Restaurants"**.
3. Find and iterate over all business elements on the current page.
4. Open the business page in a new window or tab.
5. Iterate over all pages of reviews for the current business and extract review information mentioned above. Click the "show more" elements in the page to retrieve the extended review text.
6. Close the current window and switch back to the previous one.
7. Repeat steps 4-6 for the next business element.
8. Find and click the "Next" page link if available.

Be aware that network delays can occur, leading to exceptions like ElementNotInteractableException. To address this issue, the code incorporates delays between interactions using the ***time.sleep()*** command. Additionally, a retry technique is implemented to handle any unexpected errors that may arise during the process.

You can find our code implementing the above steps on our GitHub repo. To gain some insights on the functions used:

- Selecting checkboxes:
  - **selectCheckbox1():** This function selects checkboxes for "Coffee & Tea," "Bars & Pubs," and unselects "Restaurants."
  - **selectCheckbox2():** This function is an alternative implementation for selecting checkboxes.
- Retrieving information for a specific business:
  - **retrieveInfo(review, business_name):** This function retrieves information from each review element on the page. It contains a retry mechanism to retrieve the whole text of the review by clicking the “More” button
- Navigating through review pages of a specific business:

    - **navigateReview(driver, review_data, business_name):** This function navigates through each review element on the page and calls retrieveInfo() to extract information.
    - **goToNextReviewPage(driver, review, business_name, counter):** This function goes to the next review page if available.
- Getting review information for each business:
    - **getReviewInfo(driver, review_data):** The function goes through the review pages for every business and invokes the retrieveInfo() function to obtain information about the reviews. By employing a retry mechanism with a maximum of three attempts, we reduce the chances of missing out on information due to network problems.
- Navigating to each business page:
    - **navigate(driver, businessCount, review_data, max_retries):** This function navigates to each business page and calls getReviewInfo() to extract review information.
- Navigating through multiple pages:
    - **goToPage(driver, businessCount, counter, review_data):** This function goes to the next page of businesses and calls navigate() to navigate through each business page.

By using these functions in a loop structure, you can efficiently scrape data from multiple pages.

• • •

Data Preprocessing

Before diving into the analysis of the scraped data, it is essential to preprocess the data effectively. In this tutorial, we will walk through the necessary steps to clean and prepare the data for further analysis. By following these steps, you will learn how to preprocess text data in a systematic and efficient manner. Let's get started!

1. Removing emojis from review text: Emojis are non-textual characters that do not contribute significantly to the analysis. To eliminate emojis from the review text, we will utilize the demoji library, which provides a convenient way to remove emojis from strings.

```
import demoji

def remove_emoji(string):
    return demoji.replace(string, '')
```

2. Removing non-alphanumeric characters from review text: Text data often contains non-alphanumeric characters such as punctuation and special symbols. These characters can interfere with the analysis and are typically removed. We can achieve this using regular expressions (regex) to match and remove any character that is not a letter, number, or whitespace.

```
import re

def preprocess_text(text):
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    return text
```

3. Applying stemming on words in review text: Stemming is the process of reducing words to their base or root form. It helps to unify variations of words that have the same meaning, thereby reducing the vocabulary size. We will use the Porter stemmer from the Natural Language Toolkit (NLTK) library to perform stemming on individual words.

```
from nltk.stem import PorterStemmer

stemmer = PorterStemmer()
def stem_word(word):
    return stemmer.stem(word)
```

4. Removing stop words in review text: Stop words are common words that do not carry significant meaning and are often excluded from text analysis. NLTK provides a list of commonly used stop words that we can utilize.

```
from nltk.corpus import stopwords

stop_words_english = set(stopwords.words('english'))
stop_words_greek = set(stopwords.words('greek'))
stop_words = stop_words_english.union(stop_words_greek)
stop_words = list(stop_words)
stop_words.extend(['from', 'subject', 're', 'edu', 'use','i', 'the', 'she', 'her', 'we'])
```

5. Converting review text to lowercase: Converting the review text to lowercase is a common preprocessing step in natural language processing tasks. It ensures that the same word is treated consistently regardless of its capitalization.
6. Converting review date to dateTime: In addition to preprocessing the review text, we may also need to preprocess other columns in the dataset. For example, if we for the 'Review Date' column containing dates, we might want to convert it to the appropriate data type for further analysis. We can use the pd.to_datetime() function from the pandas library to convert the 'Review Date' column to datetime format.

```
import pandas as pd

reviews['Review Date'] = pd.to_datetime(reviews['Review Date'])
```

By following these steps, you have successfully preprocessed the data and prepared it for further analysis. The overall code, combining all the preprocessing steps for review text, can be found below:

```
# Remove punctuation, stop words and emojis, apply stemming and convert to lowercase
reviews['Review Text'] = reviews['Review Text'].map(lambda x: ' '.join([
    stem_word(word) for word in word_tokenize(
        preprocess_text(
            remove_emoji(
                x.translate(str.maketrans('', '', string.punctuation)).lower()
            )
        )
    ) if word not in stop_words
]))
```

• • •

Data – Final Format

After completing the data preprocessing steps outlined earlier, the dataframe will have the following columns: Business_name, Username, Review Date, Visit Date, Review Title, Review Text and Rating.

	Business_name	Username	Review Date	Visit Date	Review Title	Review Text	Rating
0	Albeta Mediterranean Bakery	Traveler32528783809	2023-06-02	June 2023	Highly recommended	top qualiti product except custom servic tasti...	5.0
1	Albeta Mediterranean Bakery	Tourist30751701237	2023-05-31	May 2023	Breakfast	great place breakfast realli enjoy tortilla ch...	5.0
2	Albeta Mediterranean Bakery	Maria D	2023-05-10	May 2023	You have to try it	stuff help kind also beg varieti pastri best c...	5.0
3	Albeta Mediterranean Bakery	Stella K	2023-04-03	March 2023	Best bakery/quick bites	best bakeri thessaloniki highest qualiti ingre...	5.0
4	Albeta Mediterranean Bakery	vagmarga	2023-03-03	March 2023	Delicious stuff for all day long!	everyth delici food polit staff varieti food c...	5.0
...	...	...	...	...	...	...	...
1805	Iktinou Au Trottoir	drjosephgerges	2017-10-22	August 2017	a must for narrow street lover	hiden street thessaloniki cafe must coffe narr...	4.0
1806	Iktinou Au Trottoir	Orc_L	2016-08-29	August 2016	Slow service	tabl clean previou guest unattent servic waite...	2.0
1807	Iktinou Au Trottoir	LGedik	2014-09-21	NaN	Lovely	leisur place drink dont miss excel servic good...	5.0
1808	Iktinou Au Trottoir	psaldanaf\nSanta Cruz, Bolivia	2013-11-11	November 2013	Nice place to chill out	found cafebar chanc happi atmospher good servi...	5.0
1809	Iktinou Au Trottoir	NickK286	2013-02-26	February 2013	Espresso Bar Downtown.	best place good cup coffe downtown place warm ...	5.0

Data Analysis

In the following section, we present some intriguing visualizations that provide valuable insights into the data. Our objective is to address the following topics:

1. Number of Monthly Reviews

We visualize the trend of monthly reviews across all locations over time. Which month received the highest number of reviews? Is there any noticeable seasonality in the review volume?

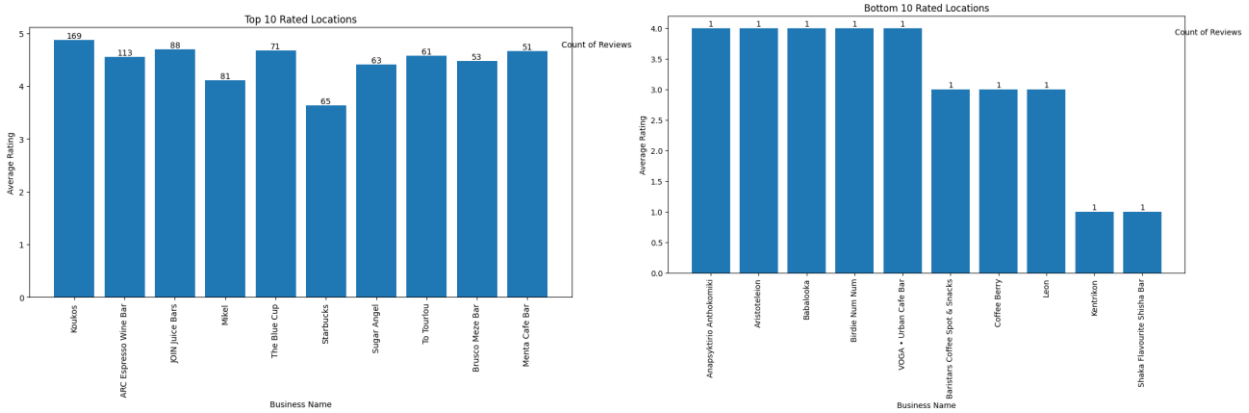
To obtain the number of monthly reviews over the years, we group the reviews by year and month and calculate the count of reviews for each month. The resulting plot indicates that the volume of reviews remains relatively consistent throughout the year. However, August emerges as the month with the highest number of reviews, suggesting a possible seasonal pattern.

2. Identifying Top-10 and Bottom-10 Rated Locations

To determine the top-10 and bottom-10 rated locations, we utilized the DataFrame's 'Business_name' column. By grouping the data based on this column, we performed calculations to obtain the sum and count of ratings for each business. Using these calculations, we derived the average rating for each business by dividing the sum of ratings by the count of reviews.

Next, we sorted the DataFrame firstly, based on the count of reviews and then based on average rating, in descending order, ensuring that businesses with higher counts and ratings appear at the top.

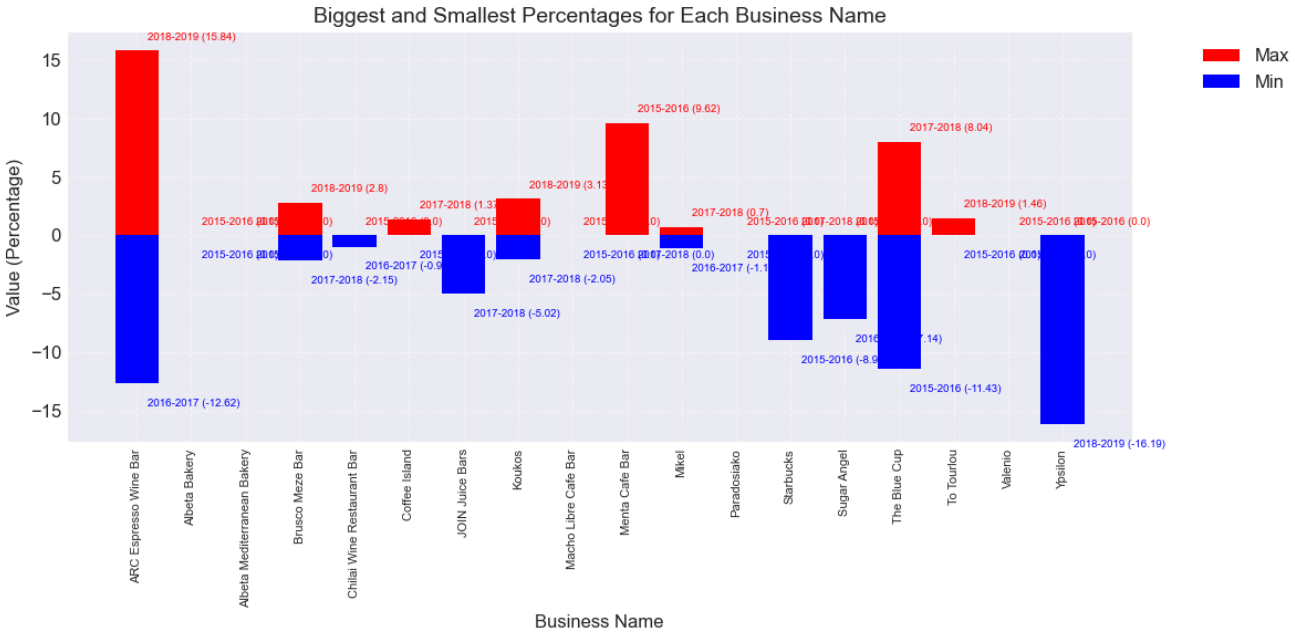
To identify the top-10 rated locations, we selected the first 10 rows from the sorted DataFrame. Similarly, to find the bottom-10 rated locations, we selected the last 10 rows from the sorted DataFrame. The following plots illustrate the Top-10 and Bottom-10 Rated businesses, their average rating and count of reviews.



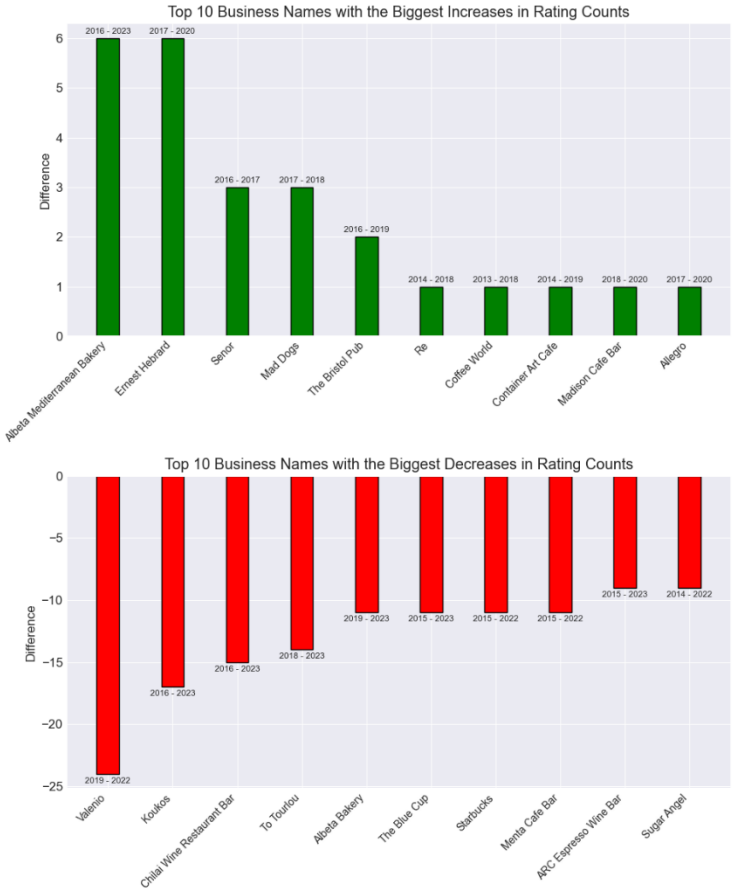
3. Identifying locations that have the highest increase or decrease in rating over the years

There are many interesting visualizations that we can show for each business. Let's see two of them.

Firstly, in order to draw better conclusions, we took only the businesses and only the years that they had a total number of ratings greater than 10. Also, for each business and for each year that it had at least 1 rating, we calculated the average rating for each year. Finally, we calculated the percentage average rating difference for consecutive years (2015-2016, 2016-2017, 2017-2018 etc.).



The first plot shows for each business with more than 10 ratings, which year they had the biggest percentage increase in the average rating, and which year they had the biggest percentage decrease.

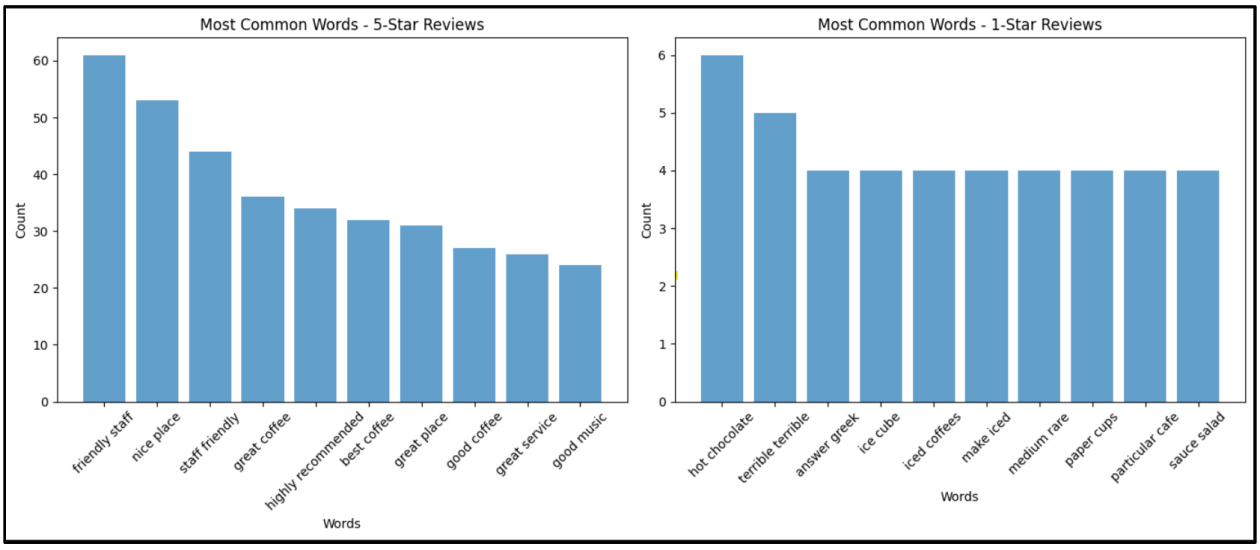


The second diagram shows the Top-10 businesses with biggest increase and decrease in the total number of ratings, taking the difference from the rating count of the year of the latest rating review and the year of the first review.

4. Visualizing Common Words, Bi-grams, and Tri-grams

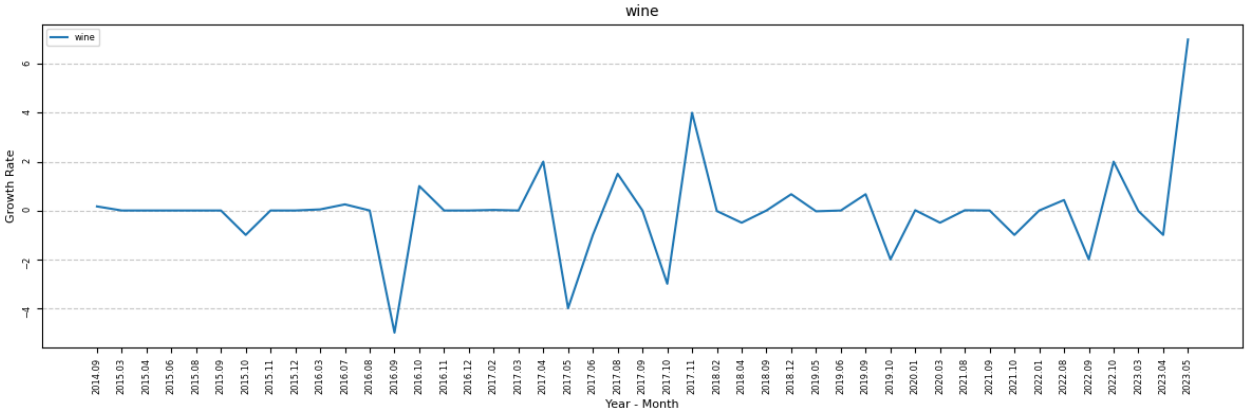
To visualize the most common n-grams, we use a vectorization technique to represent the words based on their frequency. After identifying the top 20 words, we create word cloud for most common words, bi-gram and tri-gram.

Furthermore, to compare the most common n-grams in 5-star and 1-star reviews, we follow the same process with a slight modification. We filter the dataframe to create two separate datasets, one comprising only 5-star reviews and the other containing only 1-star reviews. By applying the aforementioned steps to each dataset, we can visualize the distinct common words, bi-grams, and tri-grams in these two categories.

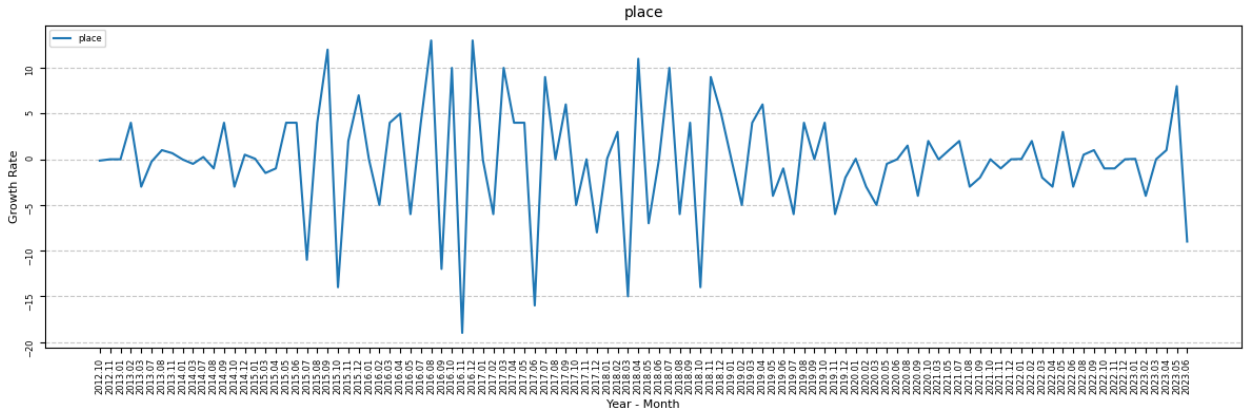


5. Visualize the 10 Fastest Growing and Shrinking Words in TripAdvisor Reviews Over Time

To understand the evolving trends in TripAdvisor reviews, we analyze word frequencies and growth rates over different months and years. By calculating the frequency of words for each month and comparing them between consecutive months, we determine the growth rate of each word. The words are then sorted based on their growth rates to identify the top 10 growing words and the top 10 shrinking words, providing valuable insights into the changing dynamics of TripAdvisor reviews. Here is an example of a fast-growing word:



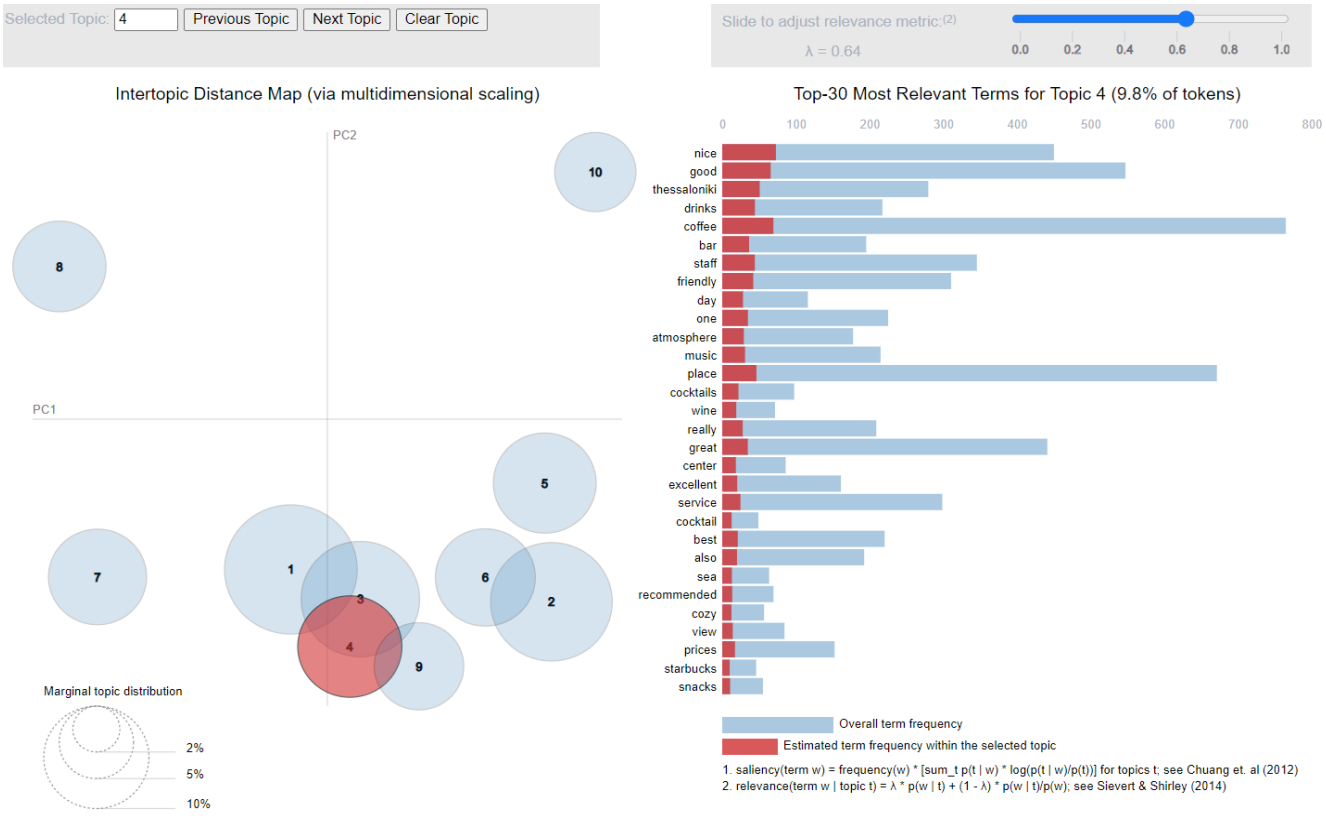
Here is an example of a fast-shrinking word:



6. Explore and visualize emerging topics from all the user reviews across time

The Intertopic Distance Map is a visualization of the topics in a two-dimensional space. The area of these topic circles is proportional to the number of words that belong to each topic across the dictionary.

The bar chart by default shows the 30 most salient terms. The bars indicate the total frequency of the term across the entire corpus. A second darker bar is also displayed over the term's total frequency that shows the topic-specific frequency of words that belong to the selected topic. If the dark bar entirely eclipses the light bar, that term nearly exclusively belongs to the selected topic.



You can learn more here.

In the above plot, we selected the topic 4, which is:

```
(4,
 '0.021*"place" + 0.016*"coffee" + 0.016*"great" + 0.011*"good" + '
 '0.009*"best" + 0.008*"thessaloniki" + 0.008*"really" + 0.008*"service" +
 '0.007*"nice" + 0.007*"staff"'),
```

We also have λ = 0.64

As we can see, the most frequent terms for this topic are the words coffee, nice, good, place, which of course makes sense, because topic 4 is one of the biggest topics and contains some of the most frequent words that we use to write a review.

• • •

Insightful conclusions

And that’s all! Now, you should now have a good understanding of how to install and set up Selenium and how to scrape the desired information by combining Selenium with Python. Also, an interesting conclusion from our research is that Selenium is an excellent tool to automate almost anything on the web like performing repetitive tasks on a site.

Furthermore, by analyzing the scraped textual data we unveiled further knowledge and produced results on our topic. Most specifically, after preprocessing the textual data, we produced some very interesting visualizations which gave us some deep understanding of them. Through sentiment analysis, we've unraveled the threads of customer opinions, uncovering valuable insights that businesses can leverage to enhance their offerings.

The world of TripAdvisor awaits your exploration, and the insights you gain may just be the catalyst for success in the realm of Bars, Pubs, and Coffee places. Cheers to your future endeavors!

I hope you find this tutorial useful. Please let us know if you have any thoughts or concerns.

Thanks for reading!