

# Navigating the Noise of Online Hotel Reviews



*Generated with Flux*

<b>Amit Halbreich</b>	20817393	amithalbreich	Amit.Halbreich@mail.huji.ac.il
<b>Omer Mushlion</b>	208271197	omer_mushlion	Omer.Mushlion@mail.huji.ac.il
<b>Omri Marom</b>	319133666	omrimar	Omri.Marom@mail.huji.ac.il

## 1. Problem Description

When planning a vacation and searching for accommodations, many of us turn to read user reviews on websites like booking.com, sifting through thousands of reviews. But how meaningful are these reviews? And what can we conclude from so many contradicting reviews?

Online hotel reviews have been the focus of research across various disciplines, including their impact on consumers choice and hotel performance, text analysis and summarization of reviews, and the motivations behind leaving reviews. Using multiple topics covered during the course, this works presents the **topic-indicativeness scores** ([3.1](#)), a metric that provides nuanced and semantically rich summary of user reviews; Then, we will show how this metric can be used in order to rerank hotels to improve recommendations based on user preferences ([3.2](#)), and to extract concise but meaningful insights from a large amount of reviews ([3.3](#)).

## 2. Data

We implemented a JavaScript tool to scrape user reviews from booking.com and packaged it into a Google Chrome extension for easy use. When running the script on the reviews page of a hotel on booking.com, it automatically sifts through all reviews pages and scrapes the reviews into a .csv file. Please refer to [Appendix I](#) for the instructions for running this scraping script with a demo video.

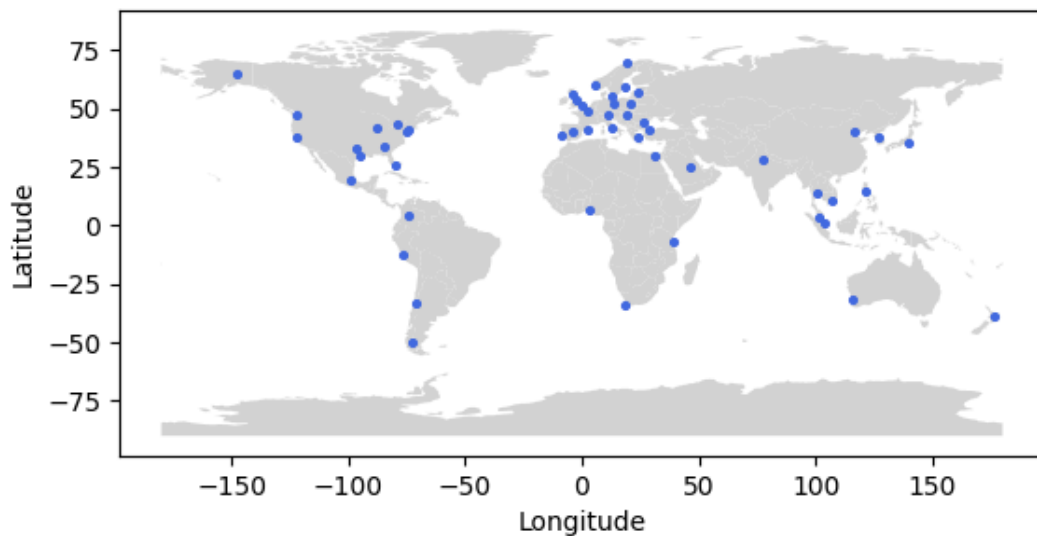
For each hotel, we get a .csv file with the following scheme:

Review Title	Negative Reviews	Positive Reviews	Rating	Stay Date	Review Date	Room Type	# of Nights	Traveller Type	Overall Average Rating
text	text	text	int {1, 10}	date mm + yy	date dd+mm +yy	categor ical	int	categoric al	float [1, 10]

We used this tool to extract the reviews of multiple accommodations with various overall ratings and accommodations types, as displayed below:

Number of 🏠 scraped	Average reviews number, per 🏠	max number of reviews of 🏠	min number of reviews of 🏠	Average overall rating of 🏠	max overall rating of 🏠	min overall rating of 🏠	Average .csv file size, in KB, per 🏠
65	1398	4059	512	8.04	9.2	3.9	502.96

The following fun bonus plot presents the locations of the hotels scrapped for this project on a world map:



Since we were looking for hotels with at least 1,000 English reviews, this map portrays the apparent bias towards the popular travel destinations in the English-speaking western world, especially among those that tend to use Booking.com. Initially, we tried to follow the list of the largest cities around the world and pick one hotel in each city, but we found out that many of those cities are either not popular travel destinations (like Karachi); reside in countries banned from Booking.com (like Tehran or Moscow); apparently do not use Booking.com much (like Beijing); or have very few English reviews, with the vast majority in Portuguese or Spanish (like São Paulo). We also wanted to pick at least one hotel from each continent, but there are no accommodations in Antarctica on Booking.com 🤖❄️

### 3. Solution

#### 3.1. Topic-Indicativeness scores of online hotel reviews

We define an online hotel review as *indicative* if it (a) repeats a point made by a *significant* portion of other reviews for the same hotel in the same sentiment and (b) does not contradict a *significant* portion of other reviews raising the same point in the opposite sentiment. For example, if only two reviews out of 200 mentions the cleanliness of the room, the conclusion about room cleanliness remains inconclusive; and if 40 reviews state that the rooms are clean while 50 reviews state that they are dirty, the matter is inconclusive as well.

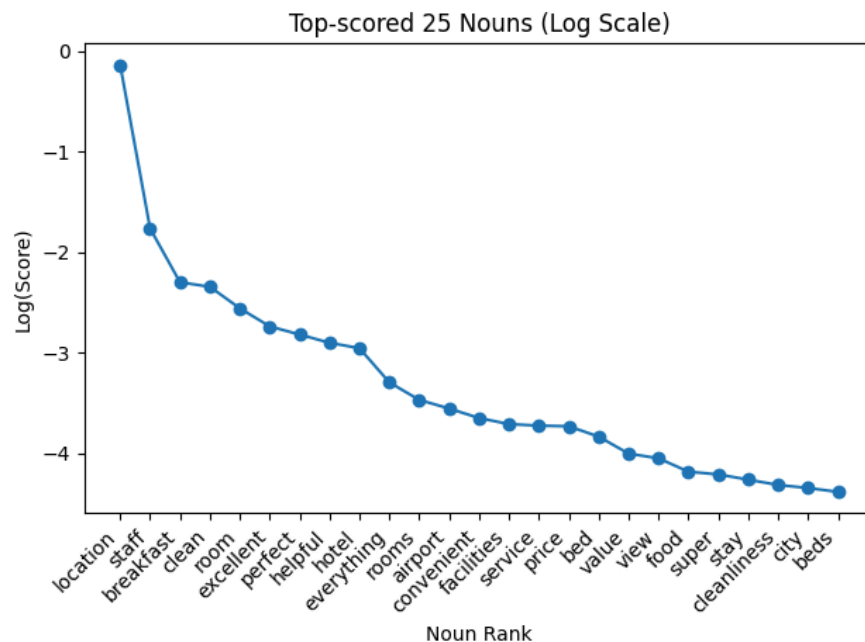
In order to do so, we need to perform three main text processing tasks: extract the top topics discussed in all reviews (3.1.1), analyse the sentiment of reviews into positive and negative reviews (3.1.2), and classify each review into one or more topics (3.1.3).

##### 3.1.1. Topic extraction from reviews

One solid, low-hanging approach to extract topics from the reviews is using an off-the-shelf topic extraction model like Bertopic. However, we chose to use tools mentioned in class:

1. **Preprocess reviews text:** tokenize to sentences, remove stop words, lemmatization.
2. **Encode each word using its TF-IDF score:** represent each word with its TF-IDF score with respect to each “document” (review sentence).
3. **Dimensionality reduction:** apply SVD decomposition to the TF-IDF matrix to reduce its dimensions while preserving as much information as possible.

The following plot presents the scores of top 25 words, filtered to include only nouns:



Based on this ranking, we grouped the top-scored words into the following 5 topics: **location**, **staff**, **food and beverages**, **room amenities** and **hotel amenities**. We can also observe the dominance of these five topics in the following word clouds of the negative and positive reviews:



### 3.1.2. Sentiment analysis of reviews

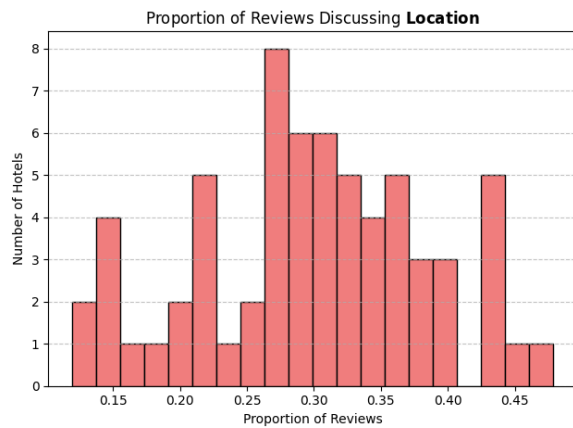
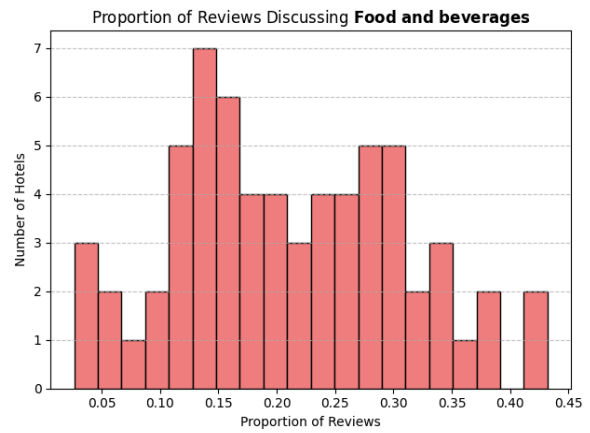
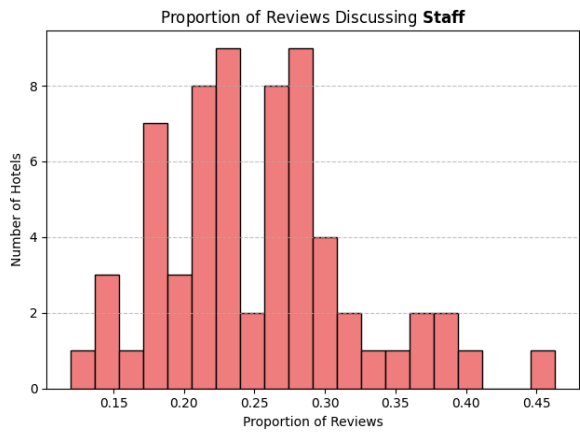
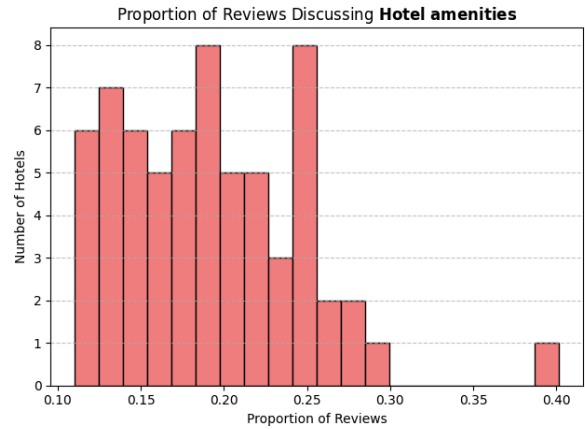
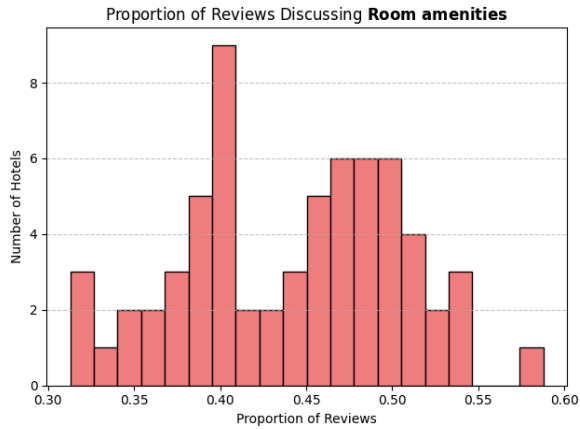
The scheme of our data already provides sentiment analysis for the reviews (separating them into positive and negative), and although it might be a bit noisy, since this is not the main topic of this project we will use this sentiment information without any additional refinement.

### 3.1.3. Classification of reviews by topics

For this task we chose to use a zero-shot text classification model with multi-label support, since many reviews discuss several topics even within a single sentence (for example: "*Staff and room cleanliness were excellent*"). We initially experimented with Facebook's `bart-large-mnli`, and later decided to use `GPT-4o` to speed up the prediction process. Please refer to [Appendix II](#) for the prompt we used for this purpose. The classification is currently binary, but exploring the use of continuous values between 0 and 1 could be an interesting direction for future work.

### 3.1.4. Topic-Indicativeness results

We mentioned two factors in the indicativeness measure we suggested. First, the following histograms present how often different topics are mentioned in reviews across various hotels:

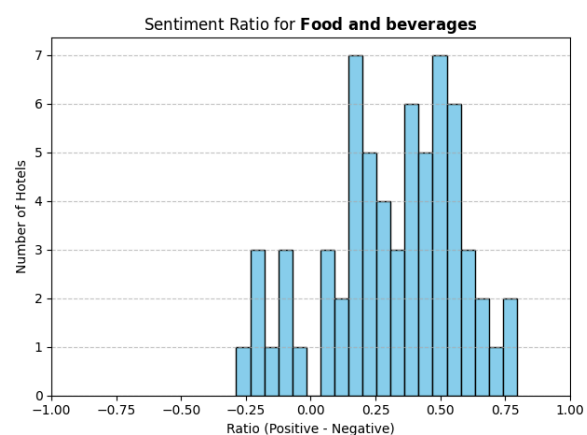
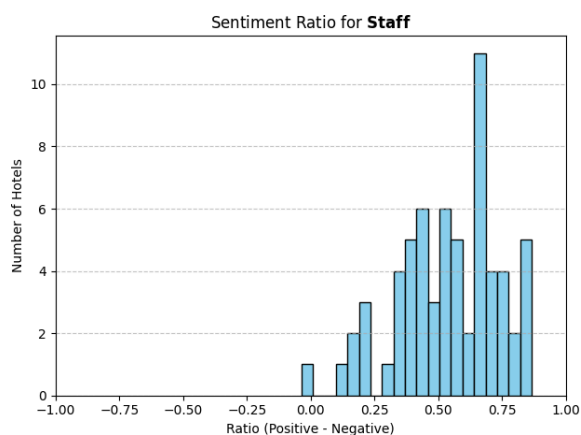
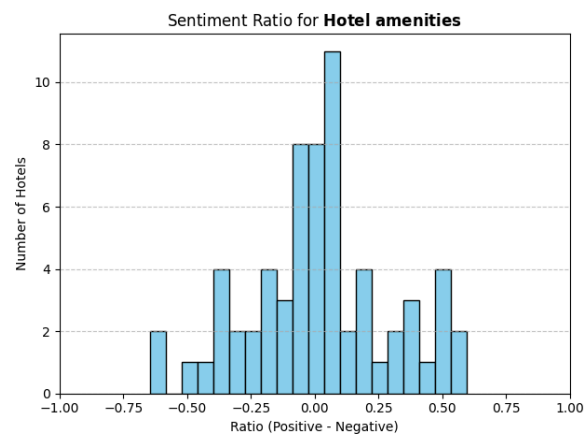
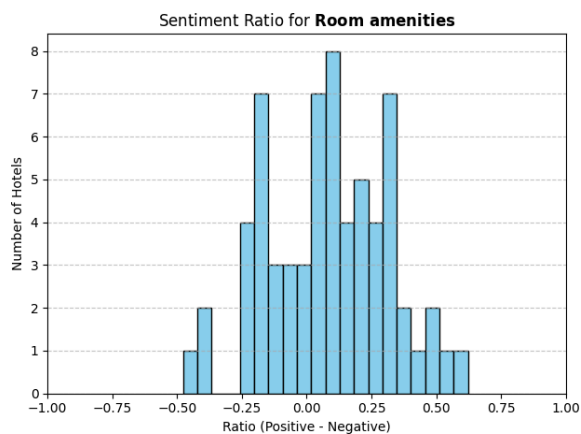


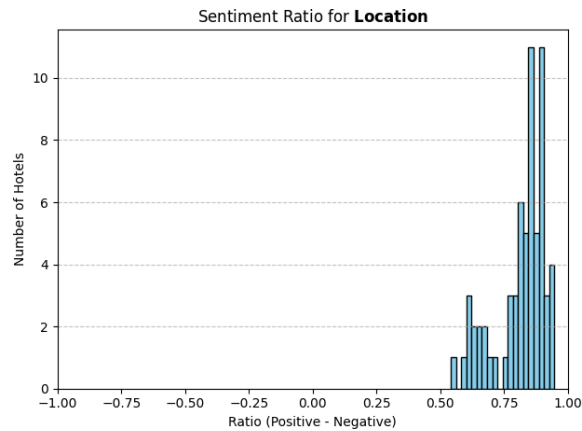
We can observe some variability in the proportion of reviews discussing different topics. This can be attributed to several factors: not all topics are relevant to every hotel (for example, budget hotels may not offer food services); there may be overlap between topics (e.g., hotel breakfast related to hotel amenities); and a notable portion of reviews are generic and do not discuss any specific topic (for example, "*Everything was good*" or "*Had a great time*"). Overall, the data shows that a substantial portion of reviews discuss these topics, confirming they are a solid basis for key topic extraction. Therefore, we will not weight the topic frequencies in our topic-indicativeness scores, although this could be explored in future work to further improve the precision and quality of the scores.

The following histograms present the balance between positive and negative reviews for each topic. This ratio is mapped to range  $[-1, 1]$ , where -1 indicates only negative reviews; 1 indicates only positive reviews; and 0 indicates an equal number of positive and negative reviews, calculated using the following formula:

$$\text{Sentiment Ratio} = (\# \text{ Positive Reviews} - \# \text{ Negative Reviews}) / (\# \text{ Total reviews for the topic})$$

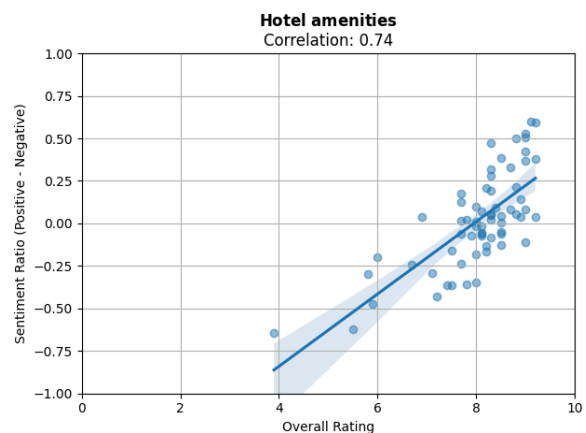
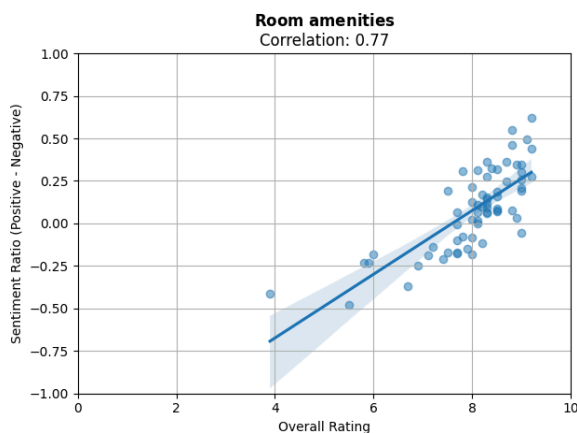
**We will use these ratios as the topic-indicativeness scores.**



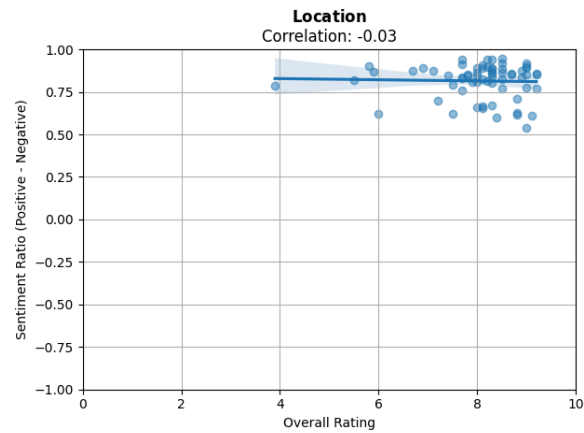
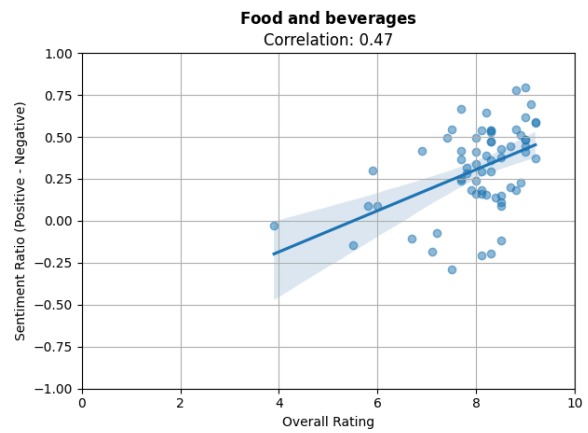
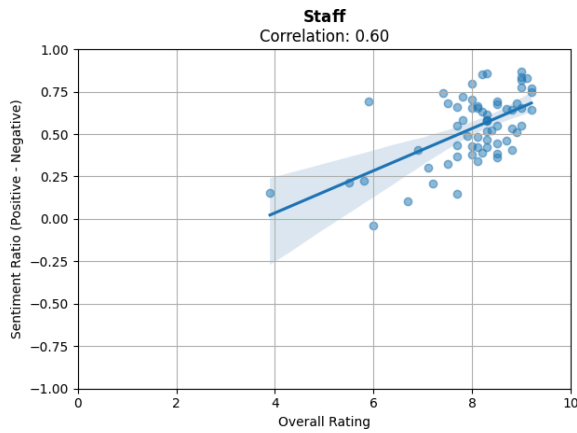


We can observe several patterns here: the sentiment ratios for "Room amenities" and "Hotel amenities" are approximately centred around 0, indicating that these topics are more debatable, with guests having mixed opinions, and people should approach reading a small subset of reviews on these topics with more caution. In contrast, "Staff" and "Location" are centred around a strong positive value, suggesting that people are more likely to agree on these topics and are generally pleased with the staff and location of the hotels. It's somewhat surprising that "Food and beverages" is also centred around a positive value (though lower than "Staff" and "Location"), as people are often quite particular about their food.

For the evaluation of these scores, we will look at the correlation between the sentiment ratios and the overall ratings of the hotels, as shown in the following plot. We want the sentiment ratios to be somewhat correlative with the overall rating, to be indicative of the hotel's rating, but not too correlative, as they should be more fine-grained than the overall rating.







Examining the Pearson correlation between sentiment ratios and overall hotel ratings, we see positive correlations with four topics, except for “Location” (which does not show any significant correlation, due to the consistently high sentiment towards location observed in the previous plot). The positive correlation of the first four topics suggests that our scores do indeed capture information about the hotel qualities, and is indicative for this.

### 3.2. Re-rank candidate hotels based on topic-indicativeness scores

We present a re-ranking or tiebreaker approach to recommend hotels to users based on the calculated topic-indicativeness results as defined in the previous section (3.1):

1. The topic-indicativeness scores serve as the item profiles for the candidate hotels.
2. We gather ratings semi-explicitly from the users by asking them to rate the importance of each of the 5 topics mentioned above (3.1.1) from 0 to 5 to their travel experience (rather than asking them to score the hotels themselves, as in pure explicit rating gathering). We use this rating as the user-profile for the recommendation purposes.
3. We calculate a weighted score of each candidate hotel based on these user ratings and the topic-indicativeness scores, to recommend hotels that better match the travel preferences of the users.




Our approach can be seen as a middle ground between collaborative recommendation approach (as it considers the ratings of other users as part of the topic-indicativeness scores calculation) and content-based recommendation (as it considers the preferences of the user itself).

Similar to the content-based recommendation approach, our re-ranking method, based on the topics importance scores, can recommend hotels to users with unique taste (for example: a user who chooses a hotel based solely on the breakfast), and it is also self-explanatory, due to the simplicity and clarity of the topic-indicativeness scores.

We will demonstrate our method as a second, re-ranking step applied to hotels and break a tie between 5 hotels in London, all with an 8.0 overall average rating (as of the data retrieval date 29/08/24). The following plot present the topic-indicativeness scores of these hotels:

<b>Hotel Name</b>	<b>Room amenities</b>	<b>Hotel amenities</b>	<b>Staff</b>	<b>Food and beverages</b>	<b>Location</b>
Euro Hotel	0.133433	0.34005	0.831029	0.563003	0.870159
Holiday Inn Express	-0.04223	0.055901	0.616438	0.201788	0.829213
Lancaster Hall	0.011655	-0.16364	0.799392	0.436735	0.876068
Merit Kensington	0.05248	-0.0765	0.139186	0.31954	0.810398
The Wellington	0.157119	-0.13402	0.744601	0.794603	0.771093

Now, consider the following three different travel characters and notice how each of them is recommended a different hotel using our approach:

Character name	 Eva Explorer	 Lucy Luxe	 Gourmet Greg
Character description	Eva's all about the thrill of the chase—exploring by day, crashing by night. She gives "Location" a 5 because she needs to be where the action is, and the rest? A solid 0. Who needs a fluffy pillow when there's a city to conquer?	Lucy's on a quest for luxury, treating every trip like a royal retreat. She gives "Hotel amenities" a 5, "Room amenities" a 4, and expects "Staff" to bow at a 4 too. The rest? A modest 1—because even a queen has limits!	For Greg, travel is all about the food. "Food and beverages" get a 5 because he's chasing every culinary delight. "Staff" is a 3—they might know the secret spots. "Location" is a 2; he'll go wherever the food is. "Room" and "Hotel amenities"? Just a 1—he only needs a place to digest.
#1 recommended 8.0 rated hotel	Lancaster Hall	Euro Hotel	The Wellington

The above character descriptions were written with the help of GPT-4o, Please refer to [Appendix II](#) for the prompt we used for this purpose.

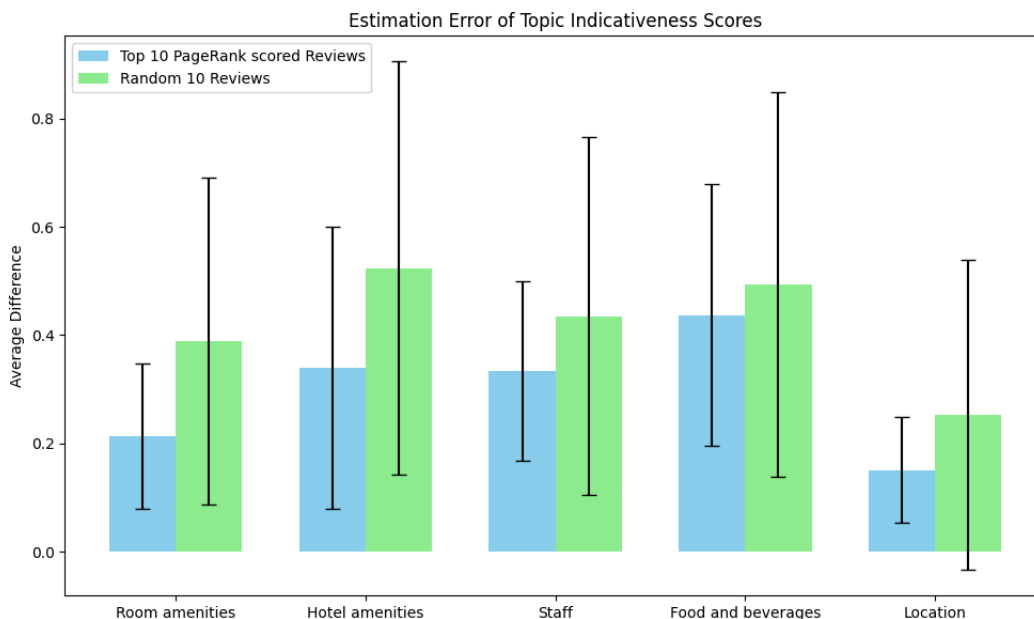
### 3.3. Extracting indicative small subset of reviews

We propose an approach to identify and extract a small, meaningful subset of reviews for each hotel that will capture the topic-indicativeness scores, so that by reading only these selected reviews, users can get a good approximation of the overall impression of the hotel.

While text summarization is a well-established research area in NLP, we aim to extract a concise yet comprehensive representation of the overall reviews using graph-based techniques:

1. For each hotel, we initialise a graph where each node represents a user review as a whole (both the positive and negative comments).
2. We calculate the “topic-sentiment vector” for each node, a binary vector of size 10 (5 topics, as defined in [3.1.1](#), and 2 sentiments) that have 1 in the  $i$ ’th coordinate iff the user review discusses the  $i$ ’th (topic, sentiment) pair.
3. Edges are added between any two nodes that share at least one common (topic, sentiment) pair.
4. The weight of each edge is determined by the normalised dot product between the “topic-sentiment vectors” of the two nodes.
5. We run PageRank algorithm on this graph to identify the most representative reviews of the overall impression of the hotel. We claim that the 10 top-scored reviews serve as a concise and indicative summary of the hotel reviews.

For evaluation, we calculated the “estimation error” of the topic-indicativeness scores using the 10 top-scored reviews, and compared it to a random subset of reviews of size 10 (averaged over multiple iterations). As presented in the following plot, this small subset of top-scored reviews provides a better indicative approximation of the topic-indicativeness scores, providing users with a concise impression of the hotel capturing the conflicting reviews and debatable points:



## Conclusion

This work deals with a challenge many of us encounter in real life: trying to obtain a conclusive impression from too many online reviews, many times with conflicting information. Apart from providing a concise, interpretable, and self-explanatory method to extract the essence of the content of the reviews (which could be easily added to such websites!), we showed several interesting and valuable applications of this measurement. As mentioned along this report, there is room for further refinement and improvement.

We hope you'll think about these things when booking your next vacation using booking.com 😊  
(Disclaimer: This work is in no way connected with, sponsored by (unfortunately...), or endorsed by booking.com or any of its affiliates.)

## Appendix I: booking.com hotel reviews scraper

Our Booking.com hotel reviews scraper is a JavaScript tool designed for automatic extraction of hotel reviews from booking.com, conveniently packaged as a Google Chrome extension for easy use.

Please click [here](#) for a must-see short cool demo video of this tool!

### Installation instructions:

1. Download or clone the project code repository to a local machine.
2. Open Google's Chrome extension management page (`chrome://extensions/`).
3. In the top right corner of the page, toggle the "Developer Mode" switch to turn it on.
4. Tap on "Load unpacked" and select the folder containing the extension files (the folder BookingHotelsReviewsScraper from the project code base).

### Using the scraper:

5. Navigate to an accommodation page on Booking.com and open the reviews page. Ensure you are viewing the English version of the page, as the scraper is designed to work with the English interface.
6. Click on the extensions icon in your browser's toolbar (typically represented by a puzzle piece icon) and select the Booking.com hotel reviews scraper extension to grant it access to the current page.
7. Click on the Booking.com hotel reviews scraper extension icon in the browser toolbar to open the extension popup interface, and use it to start scraping the hotel reviews.
8. The scraper will automatically navigate through all available review pages and extract the reviews into a .csv file.

## Appendix II: LLMs Prompts

### Classification of reviews by topics (3.1.3)

I have a dataset of hotel reviews and need your assistance in categorising them based on specific topics. The topics for analysis are:

- Room amenities: Cleanliness, condition, and features of the room (e.g., bed, bathroom, shower, air conditioning, room size, furniture).
- Hotel amenities: Cleanliness, condition, and availability of hotel facilities (e.g., swimming pool, elevator, parking, gym, lounge areas, lobby, spa, Wi-Fi, business centre).
- Staff: Interactions with hotel staff, quality of service, friendliness, helpfulness, and specific mentions of employees by name.
- Food and beverages: Food and beverage offerings (e.g., breakfast, lunch, dinner, room service, restaurant quality).
- Location: The hotel's location, proximity to attractions, convenience, views, and access to public transport.

For each topic, please add binary columns to the dataframe to indicate sentiment. For example, there should be a column for "Room amenities - positive" and "Room amenities - negative." These columns should be binary, with a value of 1 if the review (positive or negative) discusses this topic, and 0 otherwise.

Please ensure that context-aware checks are used to avoid overclassification, particularly for ambiguous keywords like "hotel" or "room" that might apply to multiple topics.

After processing the dataset according to the instructions, please provide a link to download the modified file each time I send one.

### Travel characters with a unique taste (3.2)

Create a humorous and concise description of a travel character, including their personality and preferences, along with how they would rank the importance of 'Room amenities,' 'Hotel amenities,' 'Staff,' 'Food and beverages,' and 'Location' on a scale from 0 to 5. The character should have a distinct travel style, with a unique combination of scores that reflects their priorities.