Development of a text mining tool to assess hotel reviews

Júlia Barrufet

January 11, 2021

1 Introduction

The purpose of this study is to build, using text mining techniques, an algorithm that is able to identify the key positive and negative aspects of a hotel according to the reviews given by its customers. Having this information could help hotel owners detect what are the elements they should promote in their listings to attract potential clients and which elements need to be fixed or improved.

This report describes the procedure followed to develop this text mining tool: The preprocessing of the dataset, the transformation of the documents and the three techniques employed for the analysis of positive and negative hotel reviews.

2 Preprocessing of the data

2.1 The dataset

The dataset used in this study, named "515K Hotel Reviews Data in Europe", has been obtained from the Kaggle repository[1]. This dataset contains information of 515738 reviews given by customers to various hotels in Europe. Since the developed tool would be used by individual hotels, we are only going to work with the comments of the hotel for which we have a greater number of reviews, which is the Britannia International Hotel Canary Wharf, in London.

From this dataset we are only going to use two attributes: The columns containing the postive and the negative reviews. In the following sections we describe how the dataset has been processed in order to extract this data and the process followed to prepare it in order to be properly used in our text mining application.

2.2 Preprocess the dataset

This project has been developed with the R programming language using the RStudio environment. The code of the procedures described in this report can be found as an R notebook at `https://github.com/juliabarrufet/NLP-project`. An HTML version of the R Notebook, the dataset and a README document describing how to reproduce the project can also be found in the project repository.

To apply text mining methods to our dataset we have used the `tm` library. Additional libraries used eventually can be found in the code of the project. In order to obtain a *corpus* in the adequate format to perform text mining, we followed these steps:

1. Load the data stored in a CSV file as a dataframe.
2. Get the data from the hotel with the largest number of reviews.
3. Divide the data into two dataframes, one for positive comments and one for negative comments.
4. Remove empty comments.
5. Transform the dataframes into elements of the type "Vcorpus".

[1]URL of the dataset: `https://www.kaggle.com/jiashenliu/515k-hotel-reviews-data-in-europe`

3 Processing of documents

3.1 The Term Document Matrix

The TDM can be automatically build from a corpus element and indicates the frequency of appearance of terms in a set of documents. After building this matrix we can easily obtain, for example, which are the most frequent words in the positive reviews:

```
[1] "Most frequent words in positive reviews:"
     the      and      was location     room     good    staff      for     very    hotel    great
    3058     2135     1797     1364     1267     1039      823      799      788      633      508
    were     nice    clean     with friendly     view   canary    wharf      but
     507      458      446      372      354      316      313      310      281
```

The information displayed above does not really give useful information, since the majority of the most frequent words are connectors, determinants or verbs. In the following section we will see how the documents should be processed in order to transform them into a sentence formed by the essential content of the text.

3.2 Transformation of documents

Text structures can be very complex and contain many elements that might be unnecessary for our purpose. In order to extract the most useful information from texts, we need to process them by applying the following transformations:

1. Remove punctuation.
2. Remove extra blank space.
3. Remove numbers.
4. Convert to lowercase (so Step 5 doesn't miss any term).
5. Remove stop words.
6. Stem words (deduce derived words to their word stem)

As an example, this is the result of applying the previously listed transformations to a single document:

```
[1] "Initial document:  Reasonable location in comparison to city centre Breakfast was sublime"
[1] "Transformed document: reason locat comparison citi centr breakfast sublim"
```

If we apply this list of transformations to the entire corpus and build a new TDM, these are the most frequent words in positive reviews:

```
[1] "Most frequent words in positive reviews (after applying transformations):"
     room     locat      good     staff     hotel     great      nice     clean      view    friend
     1469      1393      1039       830       660       510       466       463       417       362
      bed    canari     wharf   comfort      love     price      help breakfast      stay      valu
      333       313       310       298       285       282       271       267       263       229
```

The information displayed in this case is significantly more useful than what we obtained previously. However, many of the words still seem unnecessary for our study. Words like "hotel" and "room" will obviously be repeated in the review list of a hotel. And since we are exploring the content of positive reviews, words like "nice" or "good" will also be very frequent but do not give relevant information to understand what are the characteristics of the hotel that guests value as positive.

For this reason, we are going to add **custom stop words** to the list of words that are removed from the texts. This list of personalized stop words will include terms indicating positivity or negativity (depending on the corpus), terms related to the hotel semanthics and words that do not give relevant information but are repeated several times and distort our results.

4 Experiments and results

4.1 Study of the most frequent terms

Finally, joining this personalized lists to the standard set of stop words of the `tm` R library, we will obtain two lists of custom stop words that will be removed from each of the TDMs. After applying this transformations to the complete sets of reviews we obtain the following most frequent words:

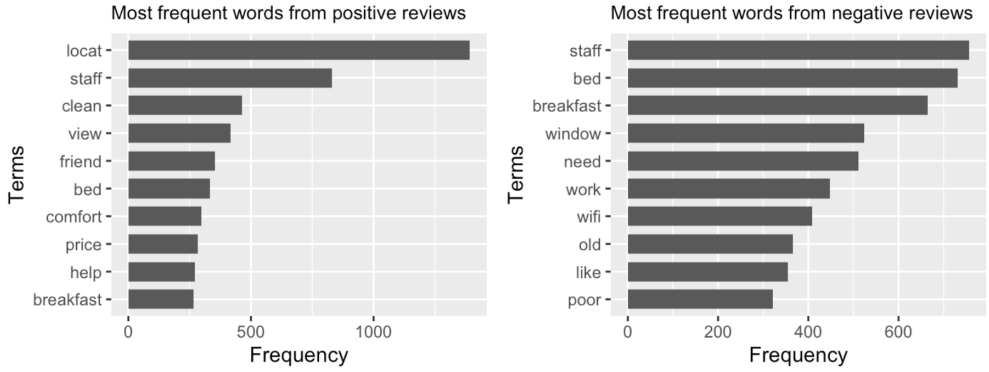

Figure 1: Histograms showing the ten most frequent words in each group of reviews.

The results shown in this graphic give an idea of the elements that customers liked or disliked from the Hotel. From this results, we can see that the location, the cleaning and the view are the characteristics of the Hotel that users valuate more positively. The beds, the breakfast and the windows are the elements for which customers have more complaints. We can easily notice that "staff" is appearing at the top of both graphics, which probably means that customers merely mention them a lot in their reviews (positive or negative). For a better understanding of situations like this we will perform other types of analysis in the following sections.

Even the results obtained in the previous section are useful, a more accurate version of the TDM can be calculate by giving weights to specific terms, resulting in a **TDM with TF-IDF weights**. In this case we assign higher weights to those words appearing in more documents (regardless of how many times a word might be repeated in a single document). Since the results obtained with the new TDM are very similar to those above, they are not reported in this document but can be visualized in the code of the project.

4.2 Associations analysis

To continue with our study, we can make an analysis of what words are more frequently associated with others. In our case we are going to explore which words have a higher association to terms that indicate both positivity and negativity. First, we build the TDM, removing the same terms as before except for those associated with positive or negative sentiments, and then we use the `findAssocs()` function to find the words that have greater correlations with the terms "good" and "bad", obtaining the following results:

	valu	breakfast	locat	price	size	choic	money	food	addit	dot	freshen	frustrat
Correlation with Good	0.2	0.19	0.15	0.15	0.15	0.13	0.13	0.12	0.11	0.11	0.11	0.11

	clearer	collag	sep	troxi	costum	leaf	stupid	advic	cafe	underneath	websit	jade
Correlation with Bad	0.28	0.28	0.28	0.28	0.2	0.2	0.2	0.19	0.16	0.16	0.16	0.15

Figure 2: Words associated more frequently with the terms "good" and "bad" in each group of reviews.

4.3 Study of the most frequent n-grams

Following the same steps as in Section 4.1, we can look for the n-grams that appear in the text with a higher frequency. This will allow us to identify compound terms that we couldn't catch in the first analysis, or characteristics that makes something be positive or negative.

First we build the corpus we will use to identify these n-grams, removing some auxiliar terms that usually appear together with others and might hinder our goal. Then we create a function that assigns tokens to n-grams (in our case, bigrams and trigrams) in order to count their appearances in the documents. Finally, as we did in Section 4.1, we plot the most frequent bigrams and trigrams of each group of reviews:

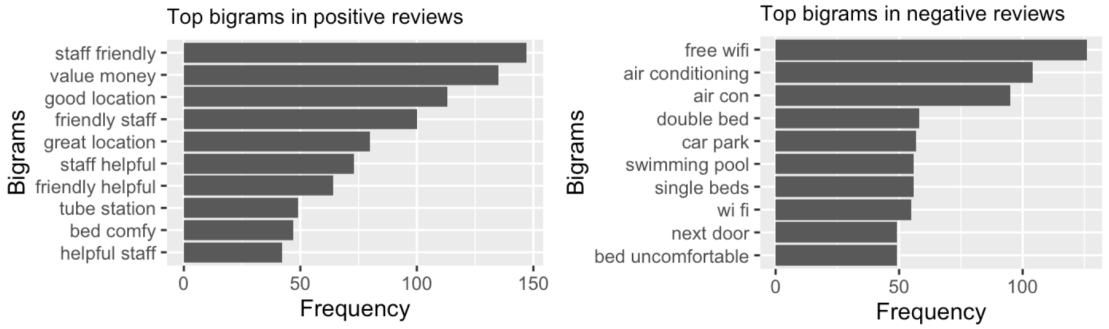

Figure 3: Most frequent bigrams in each group of reviews.

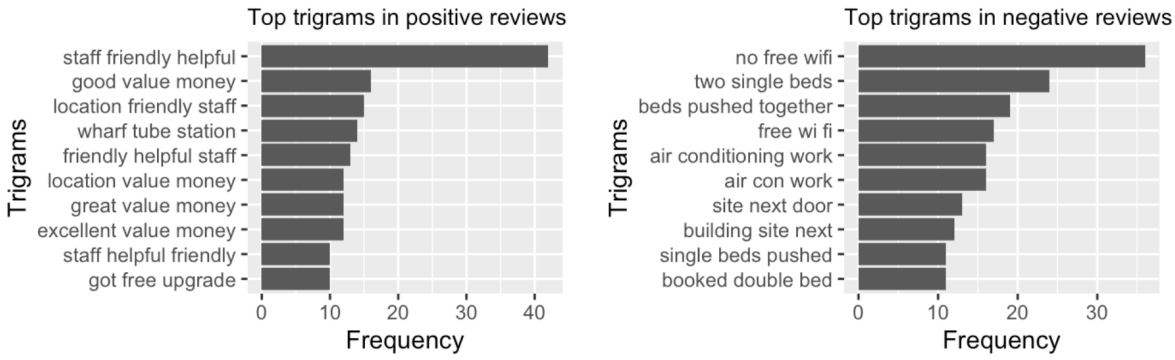

Figure 4: Most frequent trigrams in each group of reviews.

The results obtained in this section are the ones that give more useful information:

- From the study of the most frequent terms in positive reviews we saw that **location** and **staff** where the two most repeated terms. Here we can see what adjectives are most commonly found together with these terms. In summary, we can deduce that the staff is friendly and helpful and the location is great for most of the customers. We can also see that they consider the beds are comfortable and the hotel offers a good value for the money they pay.
- In the case of bad reviews, we saw that **bed**, **breakfast** and **wifi** were the terms from which we could extract more information. Through this n-grams study we can see that customers do not only think that beds are not comfortable, but they complain about the fact of finding two separate beds instead of double beds, they are disappointed because there is no free wifi and we also discovered that the air conditioning does not work well. These issues could not be appreciated in the single term study because the importance of the issue comes up when specific bigrams such as "separated beds" or "air conditioning" are put together.