UNIVERSITÀ DEGLI STUDI DI UDINE
hic sunt futura

www.uniud.it

PhD Course in Managerial and Actuarial Sciences

XXX Cycle

SOME ADVANCES IN ASPECT ANALYSIS OF USER-GENERATED CONTENT

PhD candidate
Dott. Nelu Dan

Supervisor
Dott. Alessio Fornasin

Co-supervisor
Prof. Ruggero Bellio

Year 2018

Introduction

Among many service industries, tourism industry is a typical service supplier where customers (tourists) buy already prepared service packages from tour operators or look for services themselves as in the case of the fully independent travelers (Grønflaten, 2009). The popularity of the package tours is, since many years, in decline though (Osti, 2007).

Research on social media use in tourism and hospitality proved that this relatively new tendency has significantly impacted the tourism industry (Chan & Guillet, 2011; Xiang & Gretzel, 2010), being used by tourism companies as well as by individuals. In particular, it gives to the former the opportunity to gain insight and to respond to the preferences of the consumers (Dellarocas, 2003) and to the latter the chance to share feedback or search for information (Hwang, Gretzel, Xiang, & Fesenmaier, 2006; Xiang, Gretzel, & Fesenmaier, 2009) in order to tailor a trip based on their own needs relying on other peoples' feedback (Cox, Burgess, Sellitto, & Buultjens, 2009). On one hand, numerous tourism businesses included in their websites some sort of social media services (Sanchez-Franco, 2010). On the other hand, several websites such as Tripadvisor, Trivago, Booking, gather user-generated feedback from the tourists giving anyone the opportunity to search for relevant information.

The reason why this plethora of social media and feedback gathering websites has gained popularity is that, unlike for a product evaluation, a service cannot be experienced beforehand. Thus a service is considered a high risk purchase (Babić Rosario, Sotgiu, De Valck, & Bijmolt, 2016). An experience such as a vacation that goes wrong cannot be substituted. Because of this reason, consumers do significant information research on the services they want to use (Babić Rosario et al., 2016; Moe & Trusov, 2011).

Tourism is not a mere service but a union of many service sectors (Otto & Ritchie, 1996) (transportation, accommodation, food, among others), the user's research for information spans on different qualitative dimensions for each service. For example, the accommodation service is composed by several dimensions: location, staff, sleep quality, value for the money, and so on. Services, with their dimensions allow to gather a huge quantity of information. Nowadays such information can be easily detected due to the abundance of the user-generated content in form of reviews on forums, dedicated websites or social media applications (Zeng & Gerritsen, 2014).

For several reasons, people like to share their experience (Munar & Jacobsen, 2014) giving important feedback during their experience and after it. The union of feedback sharing and the need for information from other consumers, gave birth to a mass of information pointing out critical aspects or points of strength of specific services (Gretzel & Yoo, 2008). Thus tourism becomes a good ground for opinion expression on social media.

How to evaluate this huge amount of user-generated-content is a daunting task and we propose one method to overcome some of the difficulties might arise in the information search and retrieval of products or services.

This thesis aims to investigate methods of latent rating analysis of large text corpora. Digital transformation is changing deeply how customers and companies communicate with one another. The customer-company interactions leave an increasing amount of "communication traces". The former no longer takes a mere passive role as a "performance recipient". Customers are infor-

mation deliverers, more than ever, and are interlinked through social networks, blogs, online forums or review sites. As a clear effect, they react promptly to imperfections and to positive experiences alike. This so-called user-generated content is plentiful, however the use of such information (and its integration into marketing policy planning) poses great challenges to the preparation and systematization of such *per se* unstructured texts like the reviews and the data analysis based on them. A challenge in analyzing unstructured user reviews is to make sense of the topics that are expressed by the words employed for describing their experiences and to relate them to product or service ratings. A reviewer, when describing a product, can emphasize on various aspects and neglect others. Usually these aspects are latent and statistical and computational approaches have to be employed in order to process large masses of data. Text mining and analytics are one domain that is usually employed to deal with the user-generated content. For the aforementioned reasons, reliable methods have to be studied and the present thesis tries to tackle this problem proposing a workable solution.

Starting from the online reviews associated with an overall rating, the aim is to propose a methodology for detecting the main aspects (or topics) of interest for users and to estimate the aspect ratings latently assigned in each review jointly with the weight or emphasis put on each aspect.

The thesis is structured in the form of three articles. In the first article, "Overview of Some Text Mining Tools", an overview of the current techniques applied in text mining and analytics is presented. The techniques are applied to an undisclosed dataset containing reviews related to hotel accommodations. In the beginning the opportunity to gain understanding of the text corpus with the aid of the word frequencies is evaluated. Then, an illustration on how to find group of words that compose meaningful topics is provided. The task is performed by different methods such as clustering or topic models which yield results with different degrees of precision. The tool of topic models seems to be more appropriate for our purpose, leading to the choice of applying the Latent Dirichlet Allocation (LDA) methodology.

In the second article, "Latent Aspect Rating Analysis: a Model-Based Approach", a viable proposal for estimating the aspect ratings and aspect weights is made. The existing state of the art of the related algorithms is investigated and a clearer and improved implementation is provided. The proposal is to improve on some of the current methodologies by formulating some statistically-principled alternative versions. The final result is a two-step method, which employs a suitable topics discovery in the form of sentence-based LDA at the first step, then it jointly estimates the aspect ratings and the aspect weights through a random effects Latent Rating Regression (LRR) at the second step. An application of the algorithm is provided at the end of the article.

The third article, "Evaluation and Practical Application of Latent Aspect Rating Analysis", is concerned with the application of the new algorithm and the comparison of the findings with a benchmark data set. The initial part of the article illustrates the data gathering stage, pointing out the strengths and weaknesses of the benchmark data. The remaining part of the article is devoted to the application of the two-step algorithm of the second article to the benchmark dataset. In particular, firstly the discovery of the topics using various methods is carried out. After this, by means of the proposed Latent Rating Regression approach, a set of aspect ratings and aspect weights are estimated and

compared with the benchmark data, obtaining satisfactory results. The article ends with some practical illustrations, offering some insight on the potential of the methodology for decision making in marketing research.

References

Babić Rosario, A., Sotgiu, F., De Valck, K., & Bijmolt, T. H. (2016). The effect of electronic word of mouth on sales: A meta-analytic review of platform, product, and metric factors. *Journal of Marketing Research*, *53*(3), 297–318.

Chan, N. L., & Guillet, B. D. (2011). Investigation of social media marketing: how does the hotel industry in hong kong perform in marketing on social media websites? *Journal of Travel & Tourism Marketing*, *28*(4), 345–368.

Cox, C., Burgess, S., Sellitto, C., & Buultjens, J. (2009). The role of user-generated content in tourists' travel planning behavior. *Journal of Hospitality Marketing & Management*, *18*(8), 743–764.

Dellarocas, C. (2003). The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, *49*(10), 1407–1424.

Gretzel, U., & Yoo, K. H. (2008). Use and impact of online travel reviews. *Information and communication technologies in tourism 2008*, 35–46.

Grønflaten, Ø. (2009). Predicting travelers' choice of information sources and information channels. *Journal of Travel Research*, *48*(2), 230–244.

Hwang, Y., Gretzel, U., Xiang, Z., & Fesenmaier, D. R. (2006). Information search for travel decisions. *Destination recommendation systems: Behavioral foundations and applications*, *42*(4), 357–371.

Moe, W. W., & Trusov, M. (2011). The value of social dynamics in online product ratings forums. *Journal of Marketing Research*, *48*(3), 444–456.

Munar, A. M., & Jacobsen, J. K. S. (2014). Motivations for sharing tourism experiences through social media. *Tourism Management*, *43*, 46–54.

Osti, L. (2007). *Travel guidebooks and the independent traveller in the Asia Pacific region* (PhD Thesis). Victoria University School of Applied Economics Faculty of Business and Law.

Otto, J. E., & Ritchie, J. B. (1996). The service experience in tourism. *Tourism Management*, *17*(3), 165–174.

Sanchez-Franco, M. J. (2010). Webct–the quasimoderating effect of perceived affective quality on an extending technology acceptance model. *Computers & Education*, *54*(1), 37–46.

Xiang, Z., & Gretzel, U. (2010). Role of social media in online travel information search. *Tourism Management*, *31*(2), 179–188.

Xiang, Z., Gretzel, U., & Fesenmaier, D. R. (2009). Semantic representation of tourism on the internet. *Journal of Travel Research*, *47*(4), 440–453.

Zeng, B., & Gerritsen, R. (2014). What do we know about social media in tourism? a review. *Tourism Management Perspectives*, *10*, 27–36.

Overview of Some Text Mining Tools

Abstract

The aim of this paper is to review some methods of text mining in order to extract useful information from user-generated content on the Internet. The review discusses some useful steps in text analysis. The word frequency in a corpus is an important measure of the topics discussed in it. Word association would inform the researcher on the degree of co-occurrence between co-occurrent words. Text classification into semantically significant groups is investigated through the use of two popular clustering methods, namely hierarchical clustering and k-means clustering. Unsupervised topic models are proposed as an improved alternative to clustering, trying to achieve a better word grouping under a related label name. Eventually, a supervised topic model, sLDA, is employed in order to infer topics and the ratings linked to them. Most of the treated themes are accompanied by examples.

Key words: Data scraping; LDA; R; sLDA; Text mining; Topic models.

1 Introduction

Text mining and text analytics (Fast & Elder, 2014; Miner, 2012) are techniques aimed to find and process text data in order to reveal latent connections and patterns in documents. The need for such techniques rose due to the existence of a vast amount of text information in electronic format: online newspapers, website articles, research papers, blog entries, emails, opinionated reviews and forum entries, among others. The motivation behind these techniques is to transform text into numbers so that statistical algorithms can be applied. Text data can be generated by professional or non-professional writers. In this paper we concentrate our attention on non-professional opinionated reviews. These reviews contain sentiment-laden terms (Li, Sindhwani, Ding, & Zhang, 2009). User-generated content expressed through judgments in opinionated review form on products or services became a source of information (Litvin, Goldsmith, & Pan, 2008). From the reader's point of view, review texts are very time consuming and difficult to read especially if there are many similar documents to be visually processed. It was surveyed (*PhoCusWright: "Custom Survey Research Engagement", prepared for TripAdvisor*, 2014) that 85% of the Italian `TripAdvisor` (*Tripadvisor web site*, 2017) users typically consult at least 6 - 12 reviews of a hotel in order to make a decision, giving priority to the most recent reviews. Similar considerations, of course, could be made for many other similar websites, such as Yelp, Agoda, Booking and so on; here we are also interested in other product reviews, not only related to tourism. As a consequence, extracting meaningful information from the text is a necessity and it can become a daunting task without algorithmic help. Thus, automatized techniques and statistical models must come into play and have to be used to summarize texts or to extract key aspects.

In our case, the data source is `Travel-Help`, a travel website company providing opinions on travel-related content. Due to privacy reasons and data usage policies we cannot disclose the real name of the company, the previous being fictitious. The target consisted of the English language user reviews of several *old economy* hotels situated in Boston, Massachusetts.

This paper is composed by five sections including the Introduction. Section 2 deals with the data collection issues and the structure of the obtained dataset. We extracted the digital data automatically with the aid of an open source script written in `R` (R Core Team, 2017) language. In the third section, we analyzed the downloaded reviews through common word frequency and word association text mining methods. The fourth section is dedicated to text clustering in which hierarchical and k-means clustering algorithms were employed. In the fifth section, the unsupervised topic models are presented. Topic models are about text modeling, indicating which words are probable to belong to the same subject or topic. The procedure, as input, enables us to feed text data to a computer routine and, as output, to have words assigned to topics which are relevant. The topics can be used for searching, browsing or classifying datasets. There are several models that can be used for this kind of analysis, for example Vector Space Model (VSM) (Salton, Wong, & Yang, 1975), Latent Semantic Analysis (LSA) (Landauer, Foltz, & Laham, 1998), probabilistic Latent Semantic Indexing (pLSI) (Hofmann, 1999b) or Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003).

An important variant of LDA is given by the method of Sentence-based LDA (Balikas, Amini, & Clausel, 2016; Büschken & Allenby, 2016), where each sentence entering the reviews is assigned to a single topic. Here we have followed a similar logic, so both the standard LDA where each review is treated as the basic unit of the analysis and a sentence-based LDA variant have been considered in turn. After this, we investigated the supervised learning process through the application of the supervised latent Dirichlet allocation (Mcauliffe & Blei, 2008). The procedure was applied to whole reviews as well as to sentence segmented reviews. A final consideration on the sentence-based LDA is mentioned at the end of the article and a final discussion note with future work concludes the paper.

2 Data scraping and gathering

There are several paths to obtain data sets and web scraping (Munzert, Rubba, Meißner, & Nyhuis, 2014) is one of them. It is not uncommon, for researchers in various scientific fields like economics (Cavallo, 2012), psychology (Landers, Brusso, Cavanaugh, & Collmus, 2016), agriculture (Yang, Wilson, & Wang, 2010), to obtain data through scraping. There are different methods and tools that allow data downloading. Some of them have to be constructed due to the fact that are specific to a certain field of application (see Bonifacio, Barchyn, Hugenholtz, & Kienzle, 2015), others are generic, for example `Outwit` (*Outwit web site*, 2017), `Fminer` (*Fminer web site*, 2017) or `Parsehub` (*Parsehub web site*, 2017). These methods, especially those written for a specialized purpose, frequently use computer languages as `Python` or `R` to accomplish data crawling. Through the aforementioned tools, data are transformed from unstructured to structured, making them ready for further analysis. When considering information from websites, especially the commercial ones, data is frequently the core business of the activity. Companies owning data or text data try to block or make it difficult for the automatic parsers or robots to collect information (Hirschey,

2014; Poggi, Berral, Moreno, Gavalda, & Torres, 2007; Robinson, Robinson, & Burson, 2010).

The data we are interested in is represented by the online reviews and the related metadata of the major hotels in Boston, Massachusetts. This information is resident on the touristic opinion reviews site, `Travel-Help`.

For this project, we needed a specialized crawler and we have employed the `R` language because the data to be analyzed can be organized and manipulated inside its structure. The `R` software environment allows writing routines that have crawling capabilities through some of the software packages available for this objective. Our data scraping procedure operates with three main components: a software bundle used to automatize web surfing, a stealth browser, and an R script used to download the data of interest.

The first component is `Selenium` (*Selenium web tool*, 2017), and it represents a set of various open source software tools, each one presenting a distinct action method. They are primarily used for automatizing tests concerning the functionality of websites. For our project we have used it as a remote control for a stealth browser making possible for the interface elements on web pages, for example links or pop-ups, to be accessed through script programming and not through actual mouse clicks. The tool is able to act upon `javascript` (Flanagan, 2006) procedures as well. The mentioned procedures are usually difficult to execute due to the fact that `javascript` is integrated with the browser.

The second component is the `PhantomJS` (*Phantomjs Webstack*, 2016) stealth web browser. It is a scripted, non-Graphical User Interface headless browser employed for automatizing web pages interaction. It was chosen, in lieu of a commonly used web browser like `Safari`, `Firefox` or `Internet Explorer` in order to minimize the interactions maximizing the anonymity of the user.

The last component is the downloading script. The crucial packages used for this task are: `RSelenium` (Harrison, 2014) and `rvest` (Wickham, 2015). The first ensures the interaction between `R` and `Selenium` and the second enables the actual scraping. An external file containing the links targeting the data of interest was provided. The script, sequentially starts the `Selenium` server and pauses the system for two seconds allowing it to be loaded. Then it links the driver to the `PhantomJS` browser which opens a browsing session and loads the external file. From the URL links, the name of the hotels are extracted. For each URL, the number of pages containing the reviews is calculated and, using a `for` loop, the metadata we are interested in is downloaded from the detected pages. Eventually, the data is stored in a data frame and then saved into several `csv` files, one for each targeted hotel.

The complete script for downloading travel reviews or, with slight modifications, other reviews of specific products is reported in Appendix A.

2.1 The data structure

The number of the main Hotels in Boston present at the date of scraping was 79, thus, the data, when gathered, was subdivided into the same number of files. Each file contains the reviews scraped off for the corresponding hotel. Altogether, there is a total of 93 268 separate reviews. The number of reviews is not homogeneous and it spans from 16 to 5 292 with an average of 1 180 reviews/hotel. Each review contains from a minimum of 1 to a maximum of 160 sentences with an average of 7.5 sentences/review and a median of 5. The data and meta data related to the downloaded items are: the *hotel name* which is extracted from the URL, the *id* of the review

which is a unique identifying number, the *member*'s nickname, the *quote* which represents a short summary of the review, the *date* of the review representing the temporal coordinate when the review was written, the overall *rating* and the *review* text. All the files are binded into a single file to be loaded into the `R` environment and formatted for the data analysis procedure.

2.2 Text data preparation

In text mining and analytics it is essential to normalize or preprocess data (Meyer, Hornik, & Feinerer, 2008; Miner, 2012). The function `tm_map` of the `tm` (Feinerer & Hornik, 2015) `R` package was used for this task. First of all, the text data is transformed to lowercase because a word has to be exactly the same every time it appears. After that, the punctuation has to be removed because it is useless for the analysis purpose. Numbers written in digits have to be removed too. The words with lengths greater than 30 characters (Gelbukh, 2006) were disregarded because they represent either unusual words or typing errors. For the mentioned tasks, the functions `tolower()`, `removePunctuation()`, `removeNumbers()` and `wordLengths()` were used, respectively. Another important procedure is removing the "stopwords", also called common words, that usually have no analytic value (Blake & Pratt, 2001). In every text we frequently encounter uninteresting words like: "a", "and", "also", "the", etc. By nature, these words are ubiquitous and would bias the analysis if they remained in the text. In addition to the stopwords, a small list of additional words was included. It consists of words like "hotel", "also', "day", "night", "boston" or "room", which appear regularly in this text and whose permanence does not add extra information. This extra list can be different from one text to be analyzed to another, being subjective to the application domain. White space is removed too but mainly for aesthetic reasons.

To reduce the inflectional forms, stemming (Lovins, 1968) and lemmatization (Plisson, Lavrac, & Mladenić, 2004; Toman, Tesar, & Jezek, 2006) are regularly employed but in this work only stemming is implemented. A stemming algorithm refers to a heuristic computational process that tries to uniform the words to a common root or a common stem depending if we removed the prefix or the end of the words. The hope is to achieve this goal correctly most of the times. The `tm` package uses the `SnowballC`(Bouchet-Valat, 2014) package for stemming which in turn implements Martin Porter's version of the stemming algorithm (Porter, 1980). After the normalization process, the document-term matrix (`dtm`) (Hofmann, 1999a) is created. The term `dtm` represents a matrix with two dimensions, (i, j) whose rows, i, represent the documents and the columns, j, the terms. When a term appears in a document n times, the row and column corresponding entry will have the value n, otherwise it will be 0. In our case, after the text preprocessing, we obtain 776 127 documents and 20 638 terms. The higher number of documents was obtained by the splitting of reviews into sentences. Only a few number of terms appear inside each document, thus, the result is a very sparse matrix (Tewarson, 1973). An example can be the term frequency-inverse document frequency (`tf-idf`) (Aizawa, 2003) which is widely employed in information retrieval (Forman, 2008). Each matrix value represents the frequency of the specific term "i" which appears in the document "j". The inverse document frequency measures, on the corpus, the number of documents which contain the term "i". The `tf-idf` score is the product of the two metrics: `tf` $\times$ `idf`. The `tf-idf` score increases when term "i" appears frequently in document "j" while it decreases if the term appears in other documents (Ramos, 2003). As a last operation, the sparse terms are removed from the `dtm`. These terms are words that appear very infrequently in a document and

are represented by a `dtm` that contains a huge number of entries set to zero.

The `sparse` option from the `removeSparseTerms()` function was used. The sparsity of a term refers to a limit of a relative document frequency above which the term will be discarded. Sparsity decreases as it approaches 1.0 and can take values in the open interval (0, 1.0).

For example, `sparse = 0.001` will keep the terms that have to appear in most of the documents, so they are not sparse but frequent. An example of such a word can be the article "the".

At the other side of the interval if we take, `sparse = 0.999` the function will remove only the terms that are more sparse than 0.999, thus only the very least frequent terms are discarded. An example of such a word can be "discombobulate" (Anderson & Corbett, 2017). The mathematical interpretation for `sparse = 0.999` is that, all terms for which document frequency $df_i > N \cdot (1 - 0.999)$ will be retained, where N represents the number of documents. So with a sparsity threshold of 0.999, we check for a term that occurs at least in $0.001 \times N$ documents, and, probably, most of the terms would be retained.

3 Word frequencies

After the text preprocessing, a simple analysis can be developed in order to search for words that best describe the content (Baayen, 2001; Carroll & Roeloffs, 1969). One objective is to find the frequency for each term. We calculate it by adding up the term occurrences columnwise. Then we sort them in decreasing order. As a preview of the data, we display the 20 most frequent terms together with their corresponding frequency:

```
[1] 20638
    great     staff     locat      walk      nice    servic      good
    60874     59725     56835     40118     39649     37771     37627
    clean       bed   restaur      help    friend   comfort      just
    33144     31518     30584     30324     29981     28878     26551
breakfast      well     place      park      area      back
    24547     23455     23134     23117     21883     21463
```

This simple form of text analysis can be visualized under the form of a list, like in the image above or in a more graphic way with plots or word clouds (Heimerl, Lohmann, Lange, & Ertl, 2014). Frequently, words can be associated because they occur together (Church & Hanks, 1990) and they can be represented on a graph. This type of analysis gives a hint on the structure that characterize the text.

3.1 Word barplot

In order to have a visual display of the most frequently employed words, a bar plot is a choice to be considered (Popescu, Mačutek, & Altmann, 2009). To plot all the words in the matrix is not feasible due to the huge dimension of the corpus, thus the words with a frequency greater than a threshold with an empirical chosen value of 18 000 was considered. The frequency threshold can be different from corpus to corpus, being linked to the text volume and the number of words to be displayed. It is important visualize the most frequent words because they will be the most

important words of the topics treated in the text. The words are shown in decreasing order on the x axis and their occurrence on the y axis (see Figure 1).

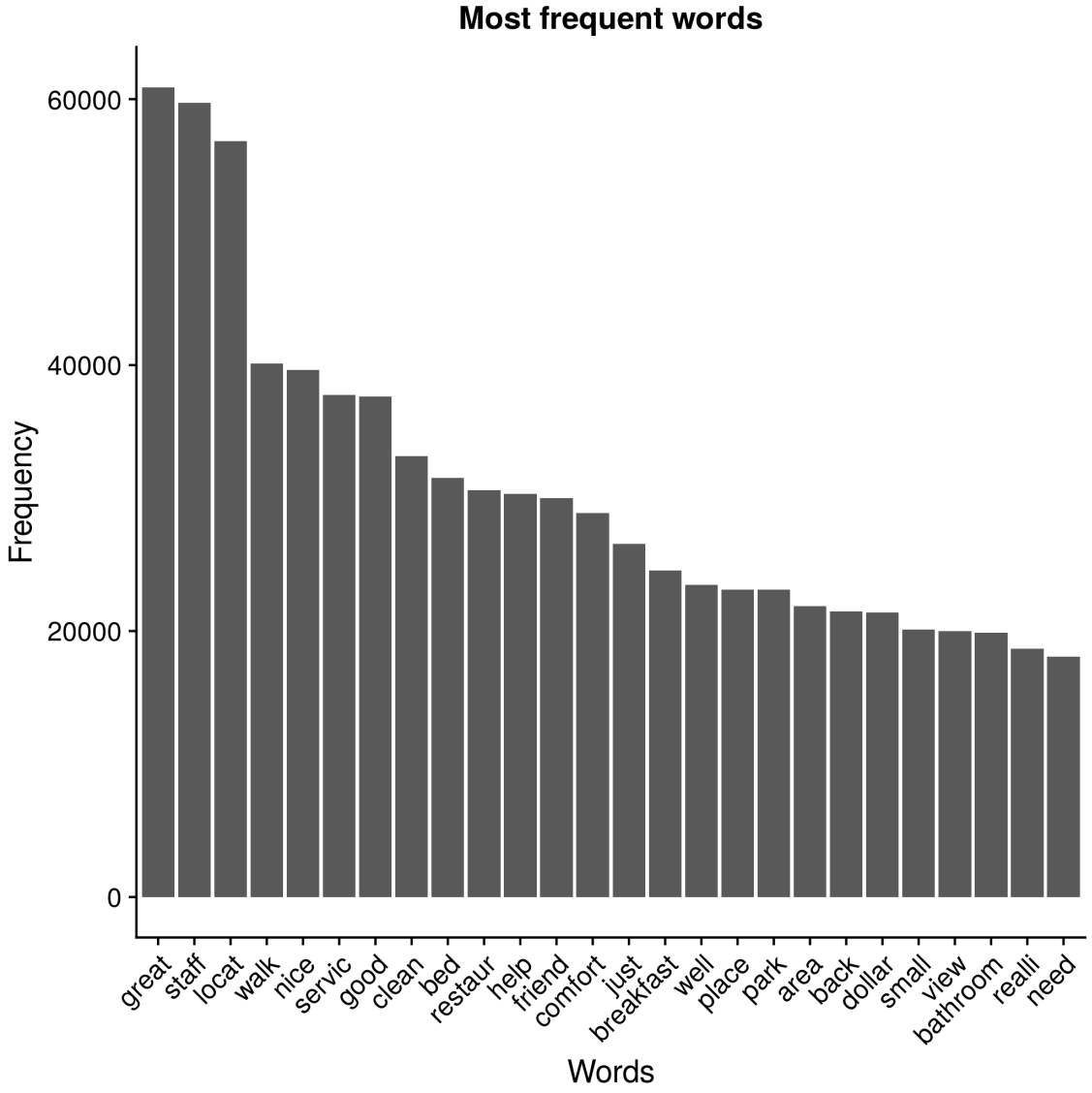

Figure 1: Most frequent words for the `Travel-Help` dataset.

It can be noticed that the nouns "staff", "locat(ion)", "servic(e)", "bed", etc. with their high frequency in the text already give a hint on what the corpus is about, these words being semantically related to accommodation amenities.

3.2 Wordcloud

A word cloud solution is also implemented to have a quick and direct visual overview of the most important words across the set of documents. It is commonly used to summarize qualitative data

(Cui et al., 2010; Scanfeld, Scanfeld, & Larson, 2010), in areas such as marketing to pinpoint important issues (Abell & Jones, 2010) or on social media analysis (Xu et al., 2013), just to mention a few applications. In this summarization procedure, huge corpora of texts are reduced to the words that appear frequently. There is a direct correlation between the words and their size appearance, bigger fonts are the words having a higher frequency, smaller fonts have consequently a lower frequency (see Figure 2). In such static representations, words are considered as stand alone entities and an association between them is not defined. The plot is obtained by the means of the `wordcloud` (Fellows, 2014) package. We notice, similarly to the previous subsection, that words like "great", "service", "staff", "walk", "locat", "nice", "restaur", "clean", "bed", "good", "comfort" have the highest frequencies. They represent the words on which the users put emphasis when writing a hotel review. Aside from the adjectives "good", "great" and "nice", which can hint positive sentiment unless accompanied by negations, the nouns reveal some topics which can be `service` or `location`. Colors group the words that have similar frequencies. Word frequency visualization is preliminary in the application of more elaborate tools for further in-depth analysis.

Figure 2: Wordcloud representation of the most frequent words in the Travel-Help dataset.

3.3 Word correlation

Words in a text are not unrelated. They are linked to other words to create meaning. To get a better grasp of our data, we can check for correlations between some of the most frequent terms and the other words. By correlation, in this context, we mean a measure of the words that co-occur in a document set. A correlation of the most frequently used words can be assessed through the `findAssocs()` function of the `tm` package. We just have to specify the document-term matrix and the correlation limit which is a number between 0 and 1. The limit represents a lower bound for the correlation strength between a chosen term and the co-occurred ones. As an example, we try to find the term co-occurrence at a correlation threshold of at least 0.10.

```
$staff
    friend       help  courteous profession     attent   accommod
      0.38       0.35       0.14       0.13       0.12       0.11

$bed
comfort    king   doubl   queen  pillow    size   comfi   linen   slept
   0.40    0.21    0.19    0.19    0.17    0.14    0.13    0.13    0.12
   sofa
   0.12

$frontdesk
call help
0.12 0.10

$locat
  great central perfect conveni    beat   ideal    walk    shop
   0.21    0.18    0.17    0.14    0.11    0.11    0.11    0.10
```

The results show connections of the word "staff" with the words "friend(ly)", "help(ful)", "curteous", "profession(al)", "attent(ive)", "accommod(ating)". The same kind of relationship is noticed for the word "bed" and the words "comfort(able)", "king", "queen", "doubl(e)", "pillow", "size", "comfi", which comes from the stemmed word "comfy", "linen", "sofa" and "slept". Similar results can be obtained for other frequently used words. The higher the number under the correlated words, the stronger the association.

3.4 Word graph

Another visual tool is a word graph. Every node is represented by a word. The co-occurrence between words is represented with a connection line. Here, as nodes, we chose the 22 most frequent words appearing more than 20 000 times in the whole corpus. We have included the links that have a correlation threshold between words of at least 0.1. The graph represents a network of associated terms built on how likely the words may appear together in a text document. This representation can be used to visualize the main relationship of co-occurrences between words (see Figure 3).

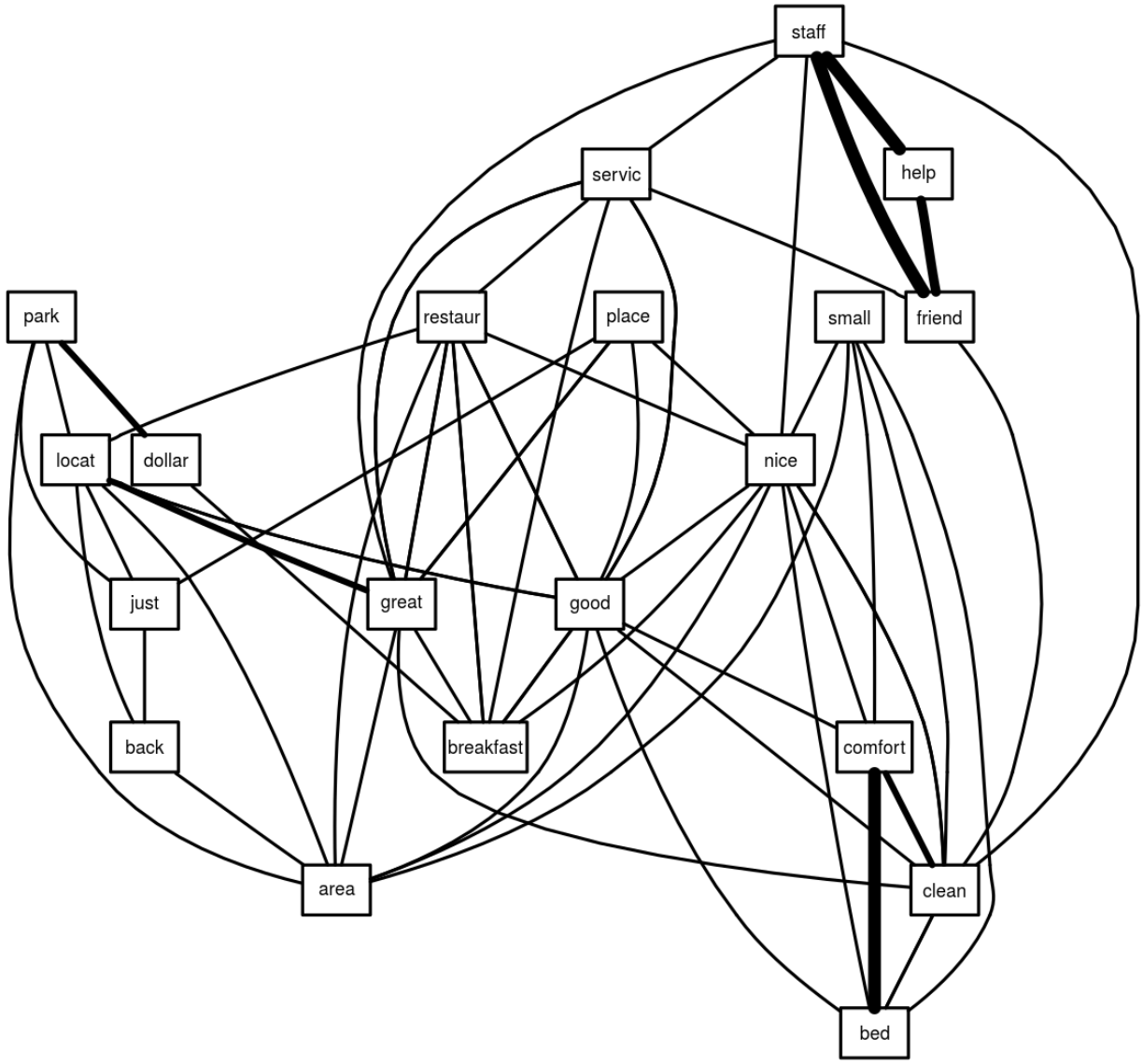

Figure 3: Representation of co-occurrences between words.

Thick lines identify a higher association, thus a stronger connection between words. The term "staff" is strongly connected to words like "friend(ly)", 0.38 or "help(ful)", 0.35 while "bed" is connected to "comfort(able)" by a association value of 0.40. The word "location" is correlated to "great" by a threshold of 0.21 and is represented by a thinner line.

3.5 Some considerations on word frequencies

The use of frequencies brings significant insight on the content of a corpus, and word associations give a better understanding of their co-occurrence in the text. Graph representations lead to a quick schematic display of the word frequencies and their association. While these methods could be adopted for a preliminary analysis of the text, they are not sufficient for discovering patterns, latent structures or grouping.

4 Clustering

A common method to study the association between terms is clustering (Jain, 2010). It comprises a set of techniques that groups objects which present similitudes but they are dissimilar to those in the other groups. In market research, clustering represents an important primary phase in segmentation studies (Arimond & Elfessi, 2001; Hruschka, 1986). In our review, grouping might ideally lead to the segmentation of the corpus into aspects. Because the similarity function is different from one method to another, there are many clustering methods that have different outcomes in the creation of groups. Although clustering does not yield an ultimate answer about grouping, it can reveal important patterns in the data. In text mining literature (Struhl, 2015), there are two types of distance based clustering algorithms that are frequently used: the `hierarchical clustering` and the `k-means`. We will tackle them with regard to word clustering in the following subsections.

4.1 Hierarchical clustering

A hierarchical clustering is an extensively studied method (Jain & Dubes, 1988) which recursively creates data clusters (Rokach & Maimon, 2005). There are many algorithms employed for hierarchical clustering and a comparison between them can be found in Zhao and Karypis (2002). For our analysis, the hierarchical agglomerative method was used because it enhances the searching process through the creation of a tree-like hierarchical structure. The general method starts with n clusters of size 1 and advances with sequential aggregations until all the values are part of a cluster. Several criteria can be applied to this kind of clustering. The Ward's method is characterized by an objective function which has to be minimized thus it produces groups, seeking to minimize the variance within them at each fusion (Murtagh & Legendre, 2014). The function `hclust` of the package `stats` which is part of the core `R` language, with `method="ward"` option, produces results corresponding to Ward's method (Ward, 1963) using a set of dissimilarities. Because we apply clustering to a `dtm`, which is encoded as a sparse matrix, there is the necessity to remove *a priori* the sparse terms as much as possible. This step is necessary to avoid an excessive word overlapping. As a consequence of this procedure, only the most frequent terms will be retained. Otherwise, it would become impossible to interpret the graphical representation. The `removeSparseTerms` function was set to 0.98, generating 34 frequent words.

After the application of the algorithm, we obtain a dendrogram graph. It has to be cut at a certain level in order to achieve the desired number of 5 clusters. This number was chosen through this exploratory approach in the pursuit of categorizing the words by aspects (see Figure 4). The aspects correspond to the topics that have ratings in `Travel-Help` and are: *Location*,

Service, *Room*, *Value*, *Experience*. Some terms are almost correctly identified: "locat(ion)" can be a self-explanatory topic, "help(ful)", "friend(ly)" and "staff" are words related to one another and belong to the *Service* topic. The "great" cluster is not well detected. Another cluster can be labeled as *Room* by the related words: "clean", "bed", "comfort". The biggest cluster encloses terms that semantically map the characteristics of *Location*, *Room* and *Value* and contains some other unrelated words. In the graph, each cluster is represented by a different color. The algorithm shows average clustering properties, further investigation has to be done in order to get better results according to our purpose of grouping words into aspects.

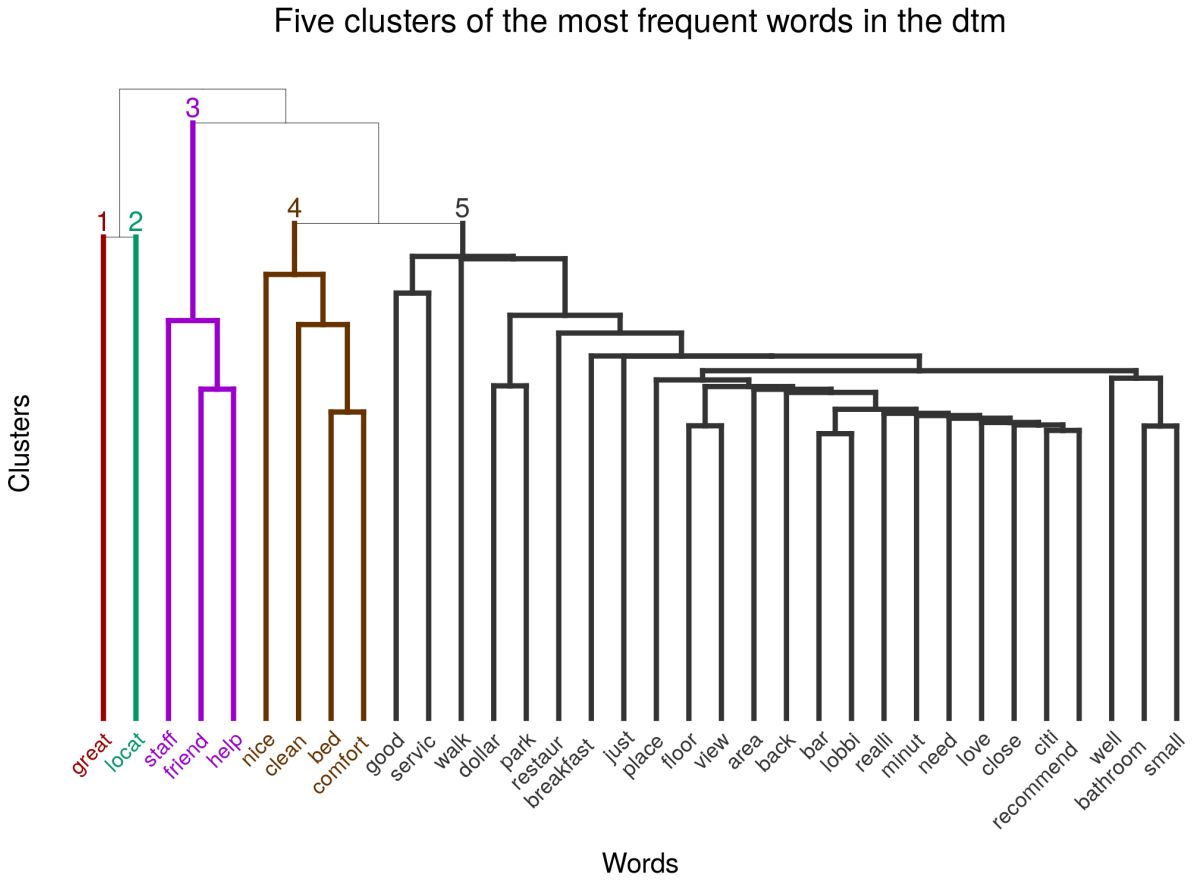

Figure 4: Agglomerative hierarchical clustering.

As a side note, the results in Figure 4 shows that the hierarchical methods of aggregation is negatively affected by the well-known chaining effect (Murtagh, 1983). One method to deal with this issue is to adopt a different clustering approach.

4.2 K-means clustering

An example of non-hierarchical clustering is the k-means method (Hartigan & Hartigan, 1975). It is a procedure to partition n objects into a predefined number k of clusters. Each object is associated to the cluster with the nearest mean in such a manner that the within-cluster distance

to the centroid is minimized. In this subsection we use a spherical k-means approach (Zhong, 2005) because it can handle sparse matrices. The difference between k-means and spherical k-means is that the first measures the Euclidean or Manhattan distance while the last uses the cosine similarity for the same purpose. For our analysis we have used the `skmeans` (Hornik, Feinerer, Kober, & Buchta, 2012) package.

The number of clusters was set to 5 for the same reasons discussed in the previous subsection. The next step was to perform the spherical k-means clustering. A necessary parameter for this kind of clustering is `m`, which is defined in the interval $[1, \infty)$. It represents the fuzziness of the cluster borders while the algorithm is build up (Kwartler, 2017). The borders get fuzzier as the parameter increases. In our case we have set the parameter to 1.2 with the intent to obtain a little amount of fuzziness. Another parameter used is `nruns` which makes the function to rerun the model the number of times desired. This procedure helps to improve the stability of the results. The type of spherical k-means applied is the `soft (fuzzy) partition` and is characterized by the fact that the cluster size is not unique. In this case, an object is assigned with a certain degree to a cluster. In the hard partition, objects belong to exactly one partition. Like in the previous section, only the most frequent words were retained, the `removeSparseTerms` function is set to 0.98 as before, yielding 34 (most frequent) words. With the aid of a barplot (see Figure 5) we can see the absolute word frequency of the clusters.

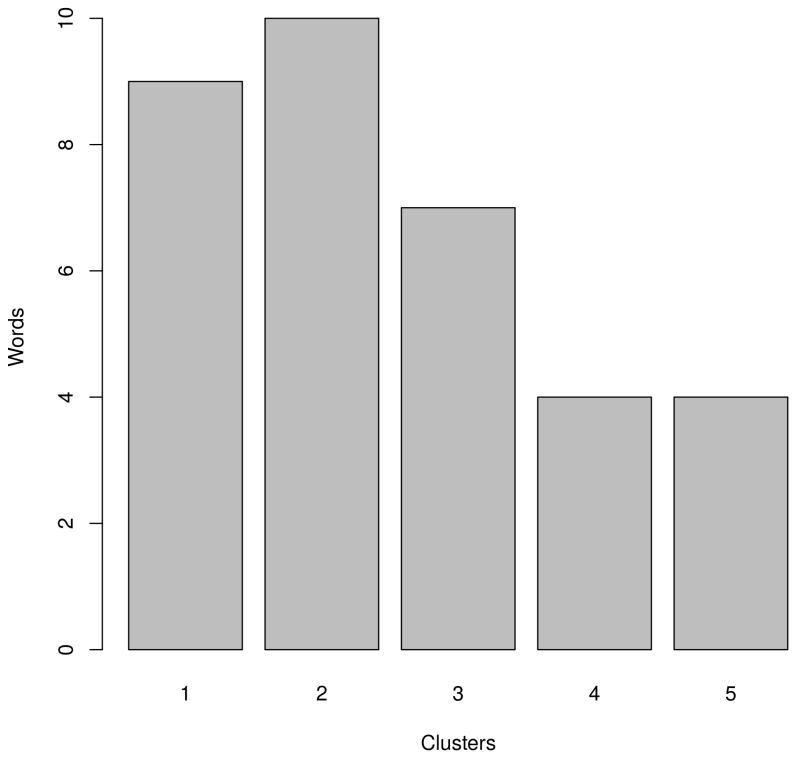

Figure 5: Topics word frequencies obtained with the application of the spherical k-means.

By the principal coordinates analysis (e.g. Kwartler, 2017), also known as multidimensional

scaling, we manage to reduce the dimensions further on, to 2 in our case. The reduced dimensions capture most of the variability between the clusters. The caption under Figure 6: "these two components explain 100% of the point variability" indicates a perfect dimensionality reduction. This is probably due to the fact that we used a small number of words.

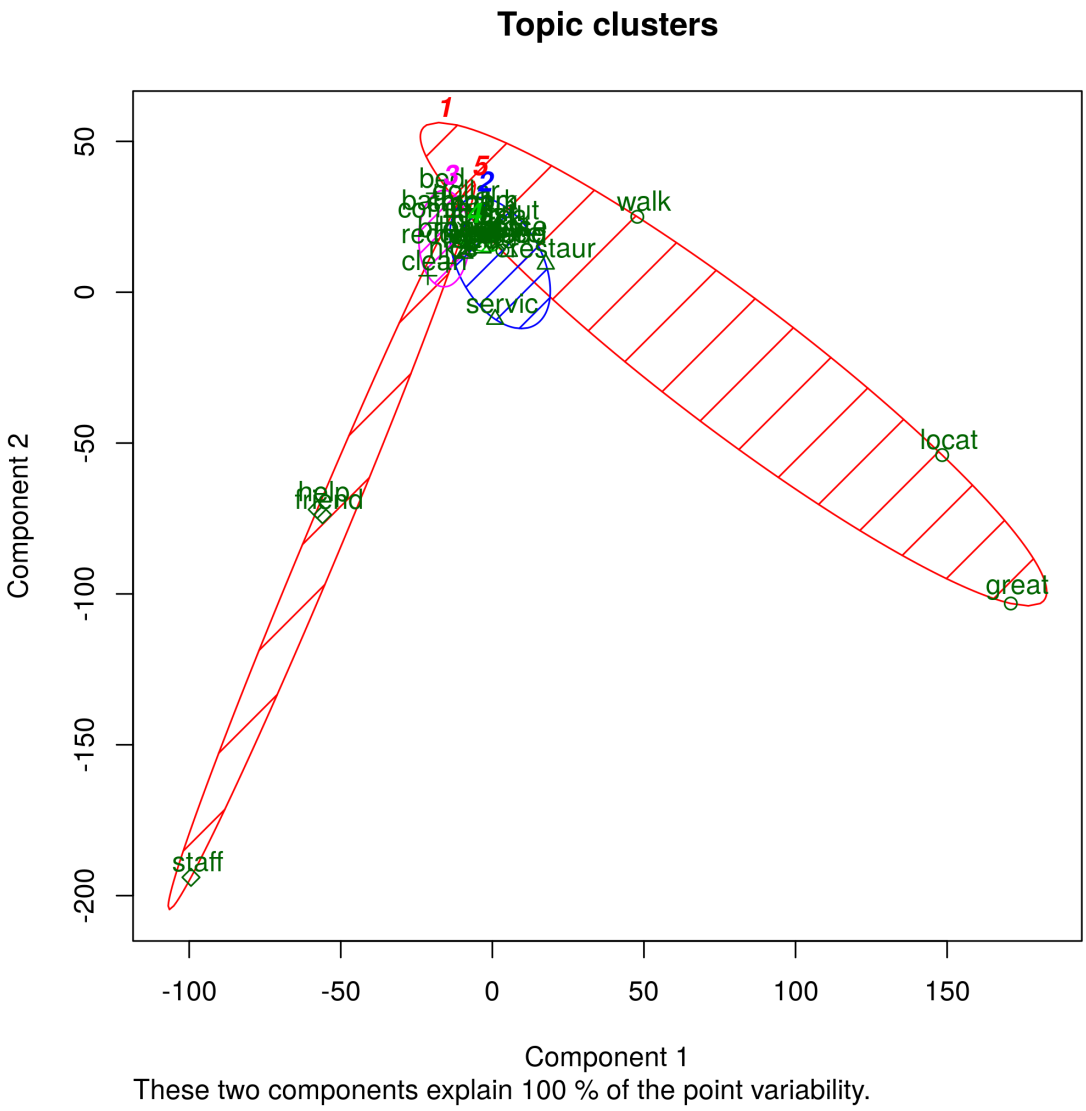

Figure 6: Topic clusters

The previous plot gives only a hint of the word clustering because the words are displayed overlapped, thus visually it is mostly unreadable. The graph shows almost all the words in an overlapping area, giving the idea of a single cluster. With the aid of the next table we can understand better how the words were grouped (see Table 1).

A silhouette plot is a way to evaluate clustering effectiveness. The completeness of the silhouette, represented by the width S_i, would indicate the definition of the cluster. Observations with a width closer to 1 are very well clustered. Values around 0 indicate the observation can be found

Table 1: Clustered words.

Topic_1	Topic_2	Topic_3	Topic_4	Topic_5
back	area	bathroom	city	friend
close	bar	bed	floor	help
great	breakfast	clean	love	need
just	dollar	comfort	view	staff
locat	good	nice		
minut	lobbi	small		
park	realli	well		
place	recommend			
walk	restaur			
	service			

between two clusters. Negative values assume the observations to be placed in the wrong cluster. In Figure 7 we notice differently colored shadows, one for each cluster.

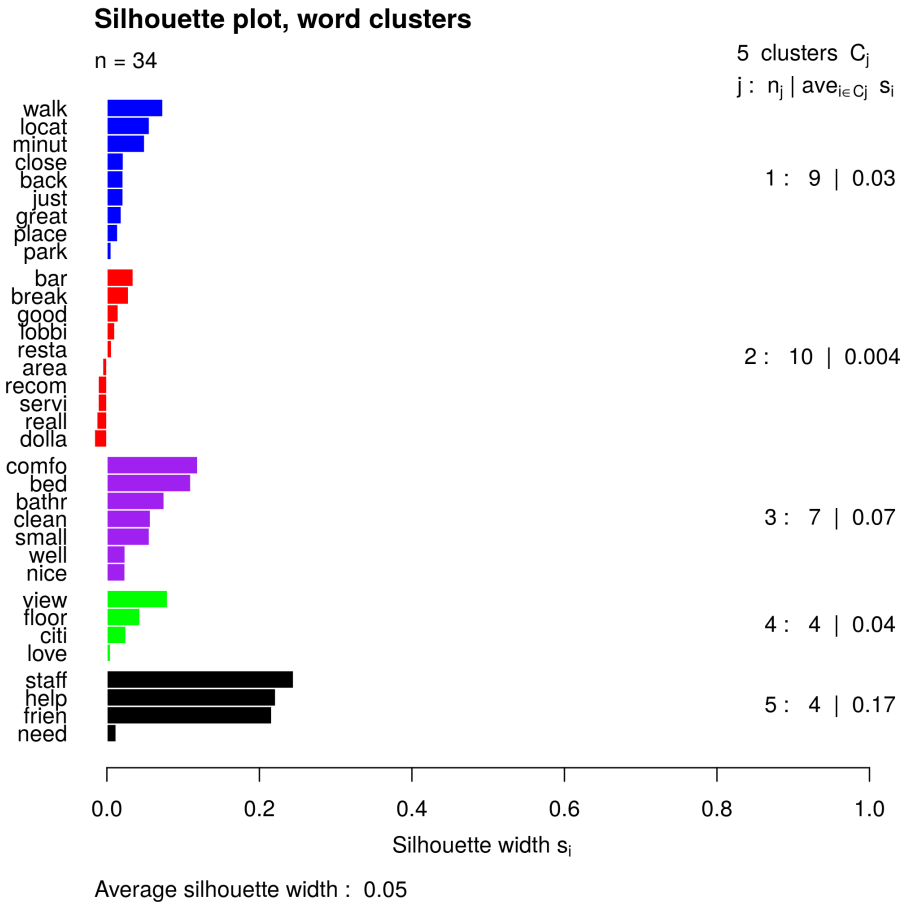

Figure 7: Silhouette of word clusters.

As it can be seen from Figure 7 and Table 1, the terms are clustered into a seemingly meaningful

way. Words like "staff", "help", "friend", "need", could indicate a *Staff* topic. Words like "walk", "locat", "minut", "close", etc. possibly indicate a *Location* topic. For instance, a *Room* topic can be represented by words like "bed", "comfort", "clean", "small", etc. Other clusterings do not seem to be so meaningful. The highest value for the best clustered words has values slightly above 0.2 which demonstrates that the clusters are not well represented.

4.3 Some considerations on clustering

Both clustering methods manage to group the words but the results do not seem to be close to our desired outcome. The objective to cluster words into aspects is only partially fulfilled because while we do have a selected number of aspects, they seem to be only partially meaningful. These unsupervised methods seem to be inappropriate in word clustering. As a consequence, an alternative approach is needed. This could be investigated through the application of topic models, more precisely through the Latent Dirichlet Allocation (Blei et al., 2003) and its subsequent developments.

5 Topic models

Topic models are "[probabilistic] latent variable models of documents that exploit the correlations among the words and latent semantic themes" (Blei & Lafferty, 2007). The main purpose behind this definition is to have an algorithm that finds themes or topics in a set of documents. This operation is useful for searching, browsing or summarizing text. The aim is to develop an unsupervised topic classification which corresponds to the aspects we are trying to identify. The procedure should automatically aggregate the words in order to form topics. A document is considered to be composed by a mixture of different topics. Topics can be defined as latent variables (having a distribution) that links words in a vocabulary and their presence in documents. The topics are assumed to be generated first, prior to the documents, and their number has to be specified in advance.

5.1 LDA

The probabilistic model we applied on the `Travel-Help` data set is the Latent Dirichlet Allocation (LDA), which has a corresponding generative process where each document is supposed to be generated as well as each word associated to a certain topic. LDA uses the Dirichlet distribution because it is the conjugate prior (Wallach, Mimno, & McCallum, 2009) of the multinomial distribution and it can be used to build an informative or non-informative prior by tuning the parameters. For a more detailed explanation on LDA see the article "Latent Aspect Rating Analysis: a Model-Based Approach" from this thesis.

5.1.1 LDA application

The LDA application to the `Travel-Help` data set needs the prior information of the number of topics, here set to 5. For our purpose, all the reviews were split into sentences (see Section 5.3). This procedure was accomplished with the aid of the `StanfordCoreNLP` (Hornik, 2017) package.

After the preprocessing of the corpus (AlSumait, Barbará, & Domeniconi, 2008), the documents and the terms convey into a document-term matrix created with the package `tm`. This matrix is submitted to the LDA procedure which is applied by the `topicmodels` (Grün & Hornik, 2011) package. By running this unsupervised function with a Bayesian approach employing the Gibbs sampling method (e.g. Darling, 2011), we obtain a particular matrix in `lda` format. The function `posterior` has to be applied in order to extract the term frequency on each topic. In our case we have extracted the top 20 keywords:

```
     topic_1   topic_2   topic_3  topic_4 topic_5
1     servic     place     staff     nice   locat
2       good recommend      help    clean   great
3  breakfast      look    friend      bed    walk
4     dollar      back      need  comfort    park
5        bar     first      make     well    area
6      lobbi      busi frontdesk    small restaur
7      excel      book     arriv     view    citi
8       free      just      made bathroom   close
9      price      next       ask    floor   minut
10      food     visit      call   beauti   right
11      much      want       way     larg  street
12    realli    experi   checkin    littl    just
13      morn      high     everi     door airport
14     water      best     check   shower  around
15     coffe    travel     guest      old perfect
16    expect    wonder      went      bit    shop
17     offer      trip      feel     suit    mani
18   restaur     enjoy     alway     work    take
19     great   definit  concierg     size     lot
20      rate    return     peopl     love    away
```

Observing the words in each topic we can semantically determine their meaning: *Value*, *Experience*, *Staff*, *Room* and *Location*. From the 20 most frequent words from each topic we can say that there is not a total correct words overlapping precisely assigned to the right topic. The majority of them do overlap though, meaning that an unsupervised classification can lead to satisfactory results when looking for aspects and their description in online reviews. The words included in the topics are coherent and it is possible to label them. In particular, Topic 1, *Value*, is described well by words like "dollar", "free", "price", "rate", "breakfast", "coffee", "food", etc. Topic 2, *Experience*, consists of a number of correctly describing words, like: "recommend", "back", "visit", "experience", "return", etc. Topic 3 as *Staff* contains many well describing words, like: "staff", "help(ful)", "friend(ly)", "frontdesk", "congierg(e)", etc. Topic 4 as *Room* holds words like "nice", "clean", "bed", "comfort", etc. Topic 5 as *Location* is endowed with a very good word description, like: "locat(ion)", "walk", "park", "area", "close", "minut(e)", etc. In literature exist automatized labeling procedures based on the ontological significance of the words within each topic (Magatti, Calegari, Ciucci, & Stella, 2009; Mei, Shen, & Zhai, 2007). The algorithm splits

the text into topics, but this is just one of the tasks we would like to accomplish. Because online reviews are characterized by a review text and an overall rating, another task would be to estimate a rating associated to each topic. This can be investigated by the Supervised LDA which is a topic model accompanied by a response variable. In the next subsection we explore its strengths and weaknesses.

5.2 Supervised LDA (sLDA)

Supervised topic models are statistical models that can be applied to labelled documents. sLDA is an algorithm which aims to analyze collections of documents in order to infer topic models and predict their response variable. The former is an additional information that can be used to guide the LDA process. For example, this model can be applied to find ratings predicted from the reviews. The difference with respect to the basic LDA algorithm is that a response variable is added to each document. sLDA models the topics and the overall rating simultaneously, finding the latent topics which would predict the response variable for a set of unlabeled documents. In this algorithm a generalized linear model is employed in order to deal with various types of response variables, such as: positive values, ordered or unordered, nonnegative integers, unconstrained real values, so on. In the case of the Travel-Help dataset, the response variable is represented by the overall rating. For a more thorough explanation on sLDA see the "Latent Aspect Rating Analysis: a Model-Based Approach" article in this thesis.

5.2.1 sLDA applied to reviews

We apply sLDA to the `Travel-Help` dataset considering the documents as reviews but not segmented into sentences. The R package `lda` (Chang, 2015) was employed for this task. Figure 8 shows the results.

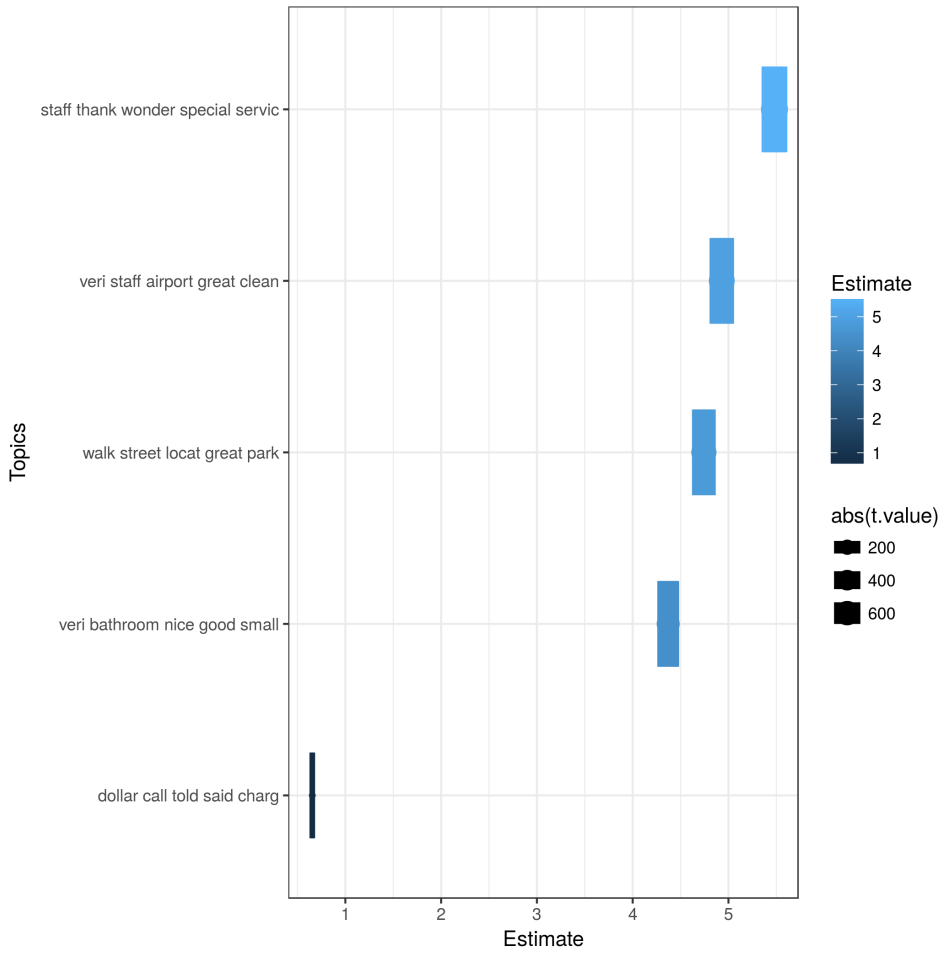

Figure 8: sLDA applied to the `Travel-Help` dataset, review based.

The plot in Figure 8 shows on the y axis the five topics, each one represented by the five most frequent words for that topic. On the x axis it is calculated the estimate of the individual rating. We can have a hint on what the computed aspects can represent. Starting from the bottom, "dollar, call, told, said, charg" could represent the *Value* aspect, "veri, bathroom, nice, good, small" could represent the *Room*. The words "walk, street, locat, great, park" would encode the *Location* aspect. The next aspect seems very ambiguous, a mixture between *Staff*, *Room* and *Location*. The last aspect, characterized by the words "staff, thank, wonder, special, service", could point to the *Staff*. The graph shows that the prediction of the rating for four topics is between 4.5 and 6. One topic is predicted to 0.5. These results would indicate high ratings towards four aspects and a negative rating towards a single one.

5.2.2 sLDA applied to sentences

Let us apply the model to the `Travel-Help` dataset after considering the subdivision of the reviews into sentences. The outcome will be significantly different. The plot in Figure 9 shows the results.

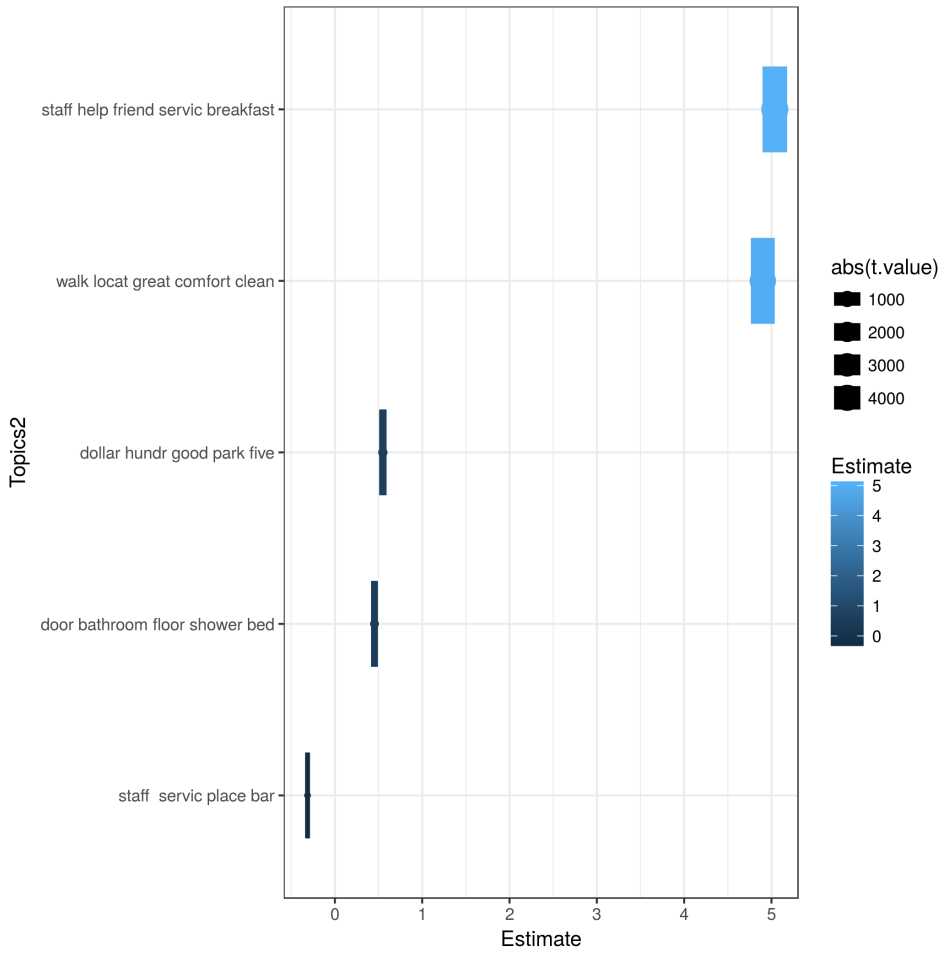

Figure 9: sLDA applied to the `Travel-Help` dataset, review based.

We notice that the outcomes are considerably distant from the previous case. The words composing the topics seem to be similar to the review based sLDA but the rating estimation is a lot different. It results that two topics are assigned to high ratings, slightly above 5 and three topics are assigned to low ratings, between -0.5 and 0.75.

5.2.3 Some considerations on sLDA

The major drawback in using sLDA is the rigid nature of the method. The aspect rating is calculated based on the word frequencies and the algorithm does not capture the sentiment polarities expressed in the text. When a certain aspect gets a positive coefficient in the sLDA model, the higher is the proportion of words belonging to that aspect employed in the document the higher will be the rating, but this approach cannot capture a real world scenario. More details on this point will be provided in the second article of this thesis. Another issue worth mentioning, though it could be considered as being of secondary importance, is the following. When we apply LDA using the Gibbs sampling estimation approach, which is by far the most commonly used approach, we obtain each time different results. This is just a special case of a general phenomenon, and

something that could be scaled down by increasing the number of MCMC simulations, yet we have found that for this model the amount of variation appears to be larger than what is usually obtained in other statistical models. In our application of the `LDA` function we used the default values for all the arguments, because otherwise for datasets as large as ours the computational time would be enormous. Since many `R` users are likely to use the default values, this seems an important limitation. We note in passing that the Variational EM (VEM) algorithm (McLachlan & Krishnan, 2007) could be used to estimate the model, since it is made available by the `LDA` function. However, we note that the results obtained with this method, although deterministic, were not satisfactory, so here we present only the results based on the Gibbs sampling, providing some evidence on the degree of variability that can be found across repeated runs of the algorithm. To this end, note that not only the topic composition changes from one run to another, but also the estimate of the rating. Sometimes, the ratings encompass the whole range of possible values, from 1 to 5 in the dataset of interest. Figure 10 summarizes the results of four runs of sLDA estimation applied to the documents corresponding to entire reviews.

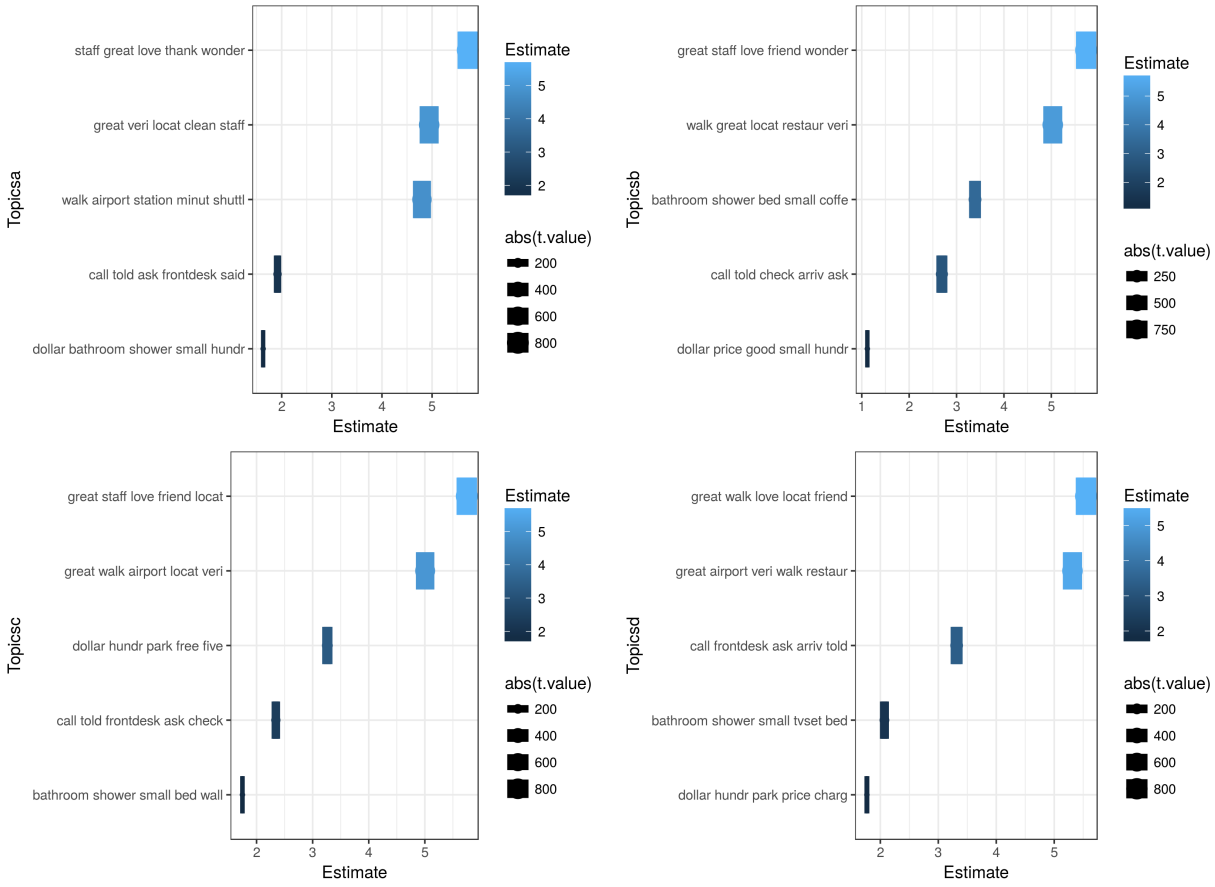

Figure 10: sLDA applied four times.

When the algorithm is applied at the sentence level, better results can be obtained. The topics are described by more stable sets of words in each graph and the rating prediction changes only slightly. There is a stable rating prediction distribution, three topics being predicted with negative ratings and two with very positive ones. There are no ratings predicted in the range 1 to 4 (see Figure 11).

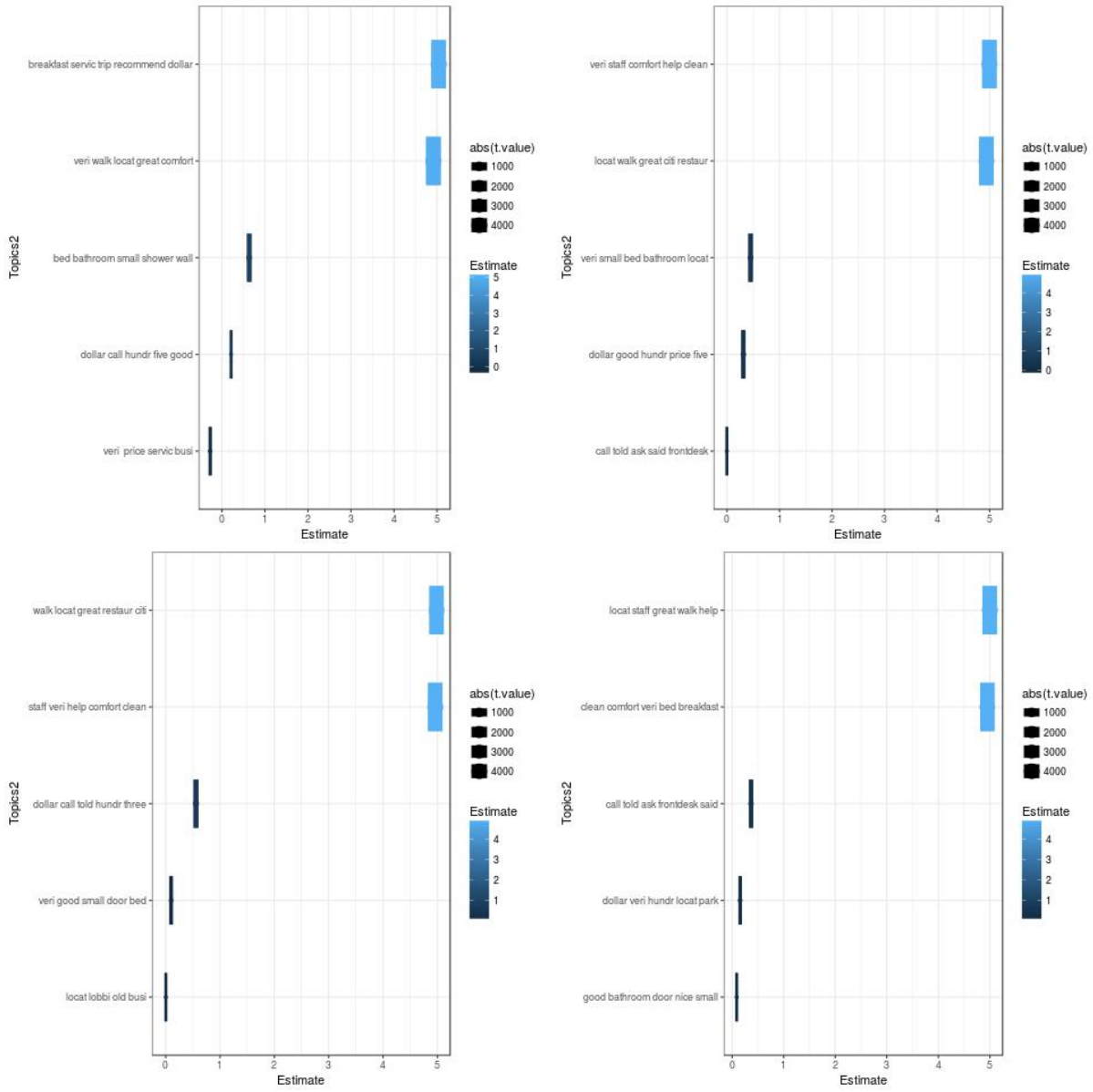

Figure 11: sLDA applied four times, sentence based.

All in all, sLDA applied at sentence level seems to be more stable regarding the topics but the rating prediction is unevenly distributed. sLDA applied at review level, seems to give unstable results as topics but more evenly distributed as ratings. Sentence-based LDA seems to yield better results than simple LDA thus we mention the algorithm in the next section.

5.3 Sentence-based LDA

In literature there are mainly two sentence-based LDA approaches. One is called Sentence-Constrained LDA (SC-LDA) (Büschken & Allenby, 2016) and the other is Sentence LDA (senLDA) (Balikas et al., 2016). They are both equivalent generative models in which persons link the sentences to a corresponding topic. This leads to the assumption that all the words in that sentence

belong to a topic. For a more thorough treatment, see the "Latent Aspect Rating Analysis: a Model-Based Approach" article from this thesis.

6 Discussion

From the previous methods we can draw the following considerations. The word frequency gives us a clear overview of the number of the most repeated terms across the reviews. An association can be found between these words and another words from the vocabulary. Clustering and topic modeling put emphasis on gathering important words under the same label. From the empirical application, we can notice that unsupervised techniques need human expert intervention in order to assess the overall quality of the results. The application of unsupervised clustering methods, like the hierarchical clustering, gives poorer results than the k-means method.

A drawback in displaying and interpreting data was the non-homogeneity of the number of the reviews and their difference in length. Another problem is the sloppy English spelling. Many words, due to the wrong spelling become unique and they are lost because of their frequency. Nevertheless, the rest of the documents, assuming they are error free, can compensate this factors. Badly written English words can increase the sparsity of the matrices introducing terms not belonging to the approved linguistic dictionary.

For what concerns the LDA application, we have noticed that the algorithm yielded better results than clustering. On the other hand, sLDA for the aim of predicting the review rating proved to be a not so reliable method, no matter if it was applied to a regular review or a sentence-based review. However we obtained slightly better results with the sentence-based version.

Through the application of the previous text mining techniques we demonstrated that results can be obtained with a variable degree of success. The application of topics models or soft clustering could empirically yield similar results, the difference consisting on how the problem is tackled and what our ultimate goal is.

7 Future work

The next step in this analysis is to expand further on the research on the metadata connected to the text. One of the most important metadata is the overall rating a user gives to the experience in a hotel structure. This score, together with the topic segmentation can be used to estimate the individual rating for each topic and the weight a user assigns to each of them. Another in-depth study can regard the time series of the overall rating using as metadata the date of the review. A comparison of the individual predicted ratings compared to collected benchmark data could be also tackled. These issues will be the subject of the remaining articles which compose this thesis.

Appendices

In this appendix we present a method used for scraping data from websites of interest. The code will require some additional adaptation for the specific website structure of interest. This

procedure should always be accompanied by the verification of the policy of the website regarding data usage and privacy.

A Appendix A

```
# load the libraries
 libs <- c("RSelenium", "dplyr","plyr","rvest","magrittr","methods")
 lapply(libs,require,character.only=TRUE)

# start selenium server (download the standalone server inside the working
    ↪ directory)
 startServer()

# gives time for the server to open up
 Sys.sleep(2)

# setup the browser type and open up the browser. download phantomjs, you can
    ↪ also use firefox
 mybrowser <- remoteDriver$new(browserName="phantomjs")
 mybrowser$open()

# load the links of the hotels (you need to change hotels_to_load with your file
    ↪ ). be sure the links are under the "link" column.
 df <- read.csv('hotels_to_load.csv')
 for (m in 1:length(df$link)){
   links <- sapply(df$link[m],as.character)

# remove duplicates if any
  links <- unique(links)

# manipulate the URL to extract information
  splitlinks <- strsplit(links, "-Reviews-")
  linksframe <- data.frame(do.call(rbind,splitlinks))
  url_0 <- links
  lefturl <- linksframe$X1
  lefturl <- paste(lefturl,'-Reviews-',sep='')
  righturl<- linksframe$X2
  righturl_no_html <- gsub('.html','',righturl)

# browse the url given
  mybrowser$navigate(url_0)
```

```
# get the number of pages (to loop through) containing the reviews
  reviews_no <- url_0 %>%
  read_html() %>%
  html_nodes("#PAGE")

  howmanypages <- reviews_no %>%
  html_nodes(".language") %>%
  html_nodes("ul>li>label span") %>%
  html_text(trim=TRUE) %>%
  as.character()

# format the pages number and transform it from character to numeric
  howmanypages <- round_any(as.numeric(gsub("\\(|\\,|\\)","",howmanypages[[1]]))
     ↪ ,10,f=floor)


# initialize variables
  website <- NULL
  list <- NULL
  lp_th_rw <- c("")

# loop through data and get the reviews
  for (i in seq(10,howmanypages,10)){
    x<-paste("or",i,"-",sep="")
    lp_th_rw <- c(lp_th_rw,x)}

# format the URL
  for(b in (lp_th_rw)){
    list[b] <- paste(lefturl,b,righturl,'#REVIEWS',sep="")}

# set the url vector
  url<-unlist(list)
  for(u in 1:length(url)){
    mybrowser$navigate(url[[u]])

    count <- c(1, 6)
# for each page, find the text
    for (j in count){
      clickfulltext <- mybrowser$findElements(using = 'css selector', value='span
         ↪ .noQuotes')

      if( length(clickfulltext) <= j )
      {clickfulltext[[j]]$clickElement()}
      else {break}
```

```
      url2 <- mybrowser$getCurrentUrl()
      url2 <- gsub("[[1]] ", "", url2, fixed=TRUE)

# parse the text structure on the page
      documents <- url2 %>%
        read_html() %>%
        html_nodes("#REVIEWS .review")

# retrieve the id of the review
      id <- documents %>%
        html_nodes(".entry p") %>%
        html_attr("id")
      id <- gsub("review_","",id)

# retrieve the quote of the review
      quote <- documents %>%
        html_nodes(".quote") %>%
        html_text(trim=TRUE)
      quote <- iconv(quote,to="ASCII//TRANSLIT")
      quote <- gsub("\"","",quote)

# retrieve the rating of the review
      rating <- documents %>%
        html_nodes(".rating_s_fill") %>%
        html_attr("alt") %>%
        gsub(" of 5 stars", "", .) %>%
        as.integer()

# retrieve the date of the review
      date <- documents %>%
        html_nodes(".rating .ratingDate") %>%
        html_text(trim=TRUE)%>%
        as.character()

      date <- gsub("\\,|\n|Reviewed |NEW", "", date)

# retrieve the member's nickname
      member <- lapply(documents, function(pn) {
        pn %>% html_node(".username") %>% html_text(trim=TRUE) %>%
          ifelse(identical(., character(0)), NA, .)
      })
      member <- unlist(member)
```

```
# retrieve the review
      review <- documents %>%
        html_nodes(".entry") %>%
        html_text(trim=TRUE) %>%
        as.character()

# save the data into a data frame
      myrow <- data.frame(id, member, quote, date, rating, review,
          ↪ stringsAsFactors = FALSE)
      myrow <- as.data.frame(myrow)
      website <- rbind(myrow, website)
      website <- unique(website)

# send the browser back and repeat the procedure
      mybrowser$goBack()}
  }
  website <- cbind(hotel_name = righturl_no_html, website)

# save the data to be furtherly processed
  save(website, file = paste(righturl_no_html, ".rda",sep=""))
  write.csv(website, file = paste(righturl_no_html, ".csv",sep=""))
  rm(list= ls()[!(ls() %in% c('mybrowser','df'))])

}

# remove the variables
  rm(list=ls())
```

References

Abell, C. H., & Jones, M. S. (2010). The power of a "Word Cloud" in marketing a nursing program. *Nursing Faculty Publications*, 40.

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, *39*(1), 45–65.

AlSumait, L., Barbará, D., & Domeniconi, C. (2008). On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Data mining, 2008. icdm'08. Eighth IEEE International Conference on* (pp. 3–12).

Anderson, W., & Corbett, J. (2017). *Exploring English with Online Corpora*. Palgrave MacMillan New York.

Arimond, G., & Elfessi, A. (2001). A clustering method for categorical data in tourism market segmentation research. *Journal of Travel Research*, *39*(4), 391–397.

Baayen, R. H. (2001). *Word Frequency Distributions*. Springer New York.

Balikas, G., Amini, M.-R., & Clausel, M. (2016). On a topic model for sentences. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016* (pp. 921–924).

Blake, C., & Pratt, W. (2001). Better rules, fewer features: a semantic approach to selecting features from text. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on* (pp. 59–66).

Blei, D. M., & Lafferty, J. D. (2007). A correlated topic model of science. *The Annals of Applied Statistics*, *1*(1), 17-35.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Bonifacio, C., Barchyn, T. E., Hugenholtz, C. H., & Kienzle, S. W. (2015). CCDST: A free Canadian climate data scraping tool. *Computers & Geosciences*, *75*, 13–16.

Bouchet-Valat, M. (2014). SnowballC: Snowball stemmers based on the C libstemmer UTF-8 library [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=SnowballC` (R package version 0.5.1)

Büschken, J., & Allenby, G. M. (2016). Sentence-based text analysis for customer reviews. *Marketing Science*, *35*(6), 953–975.

Carroll, J. M., & Roeloffs, R. (1969). Computer selection of keywords using word-frequency analysis. *Journal of the Association for Information Science and Technology*, *20*(3), 227–233.

Cavallo, A. (2012). Scraped Data and Sticky Prices. *MIT Sloan Research Papers Series*(4976).

Chang, J. (2015). LDA: Collapsed Gibbs Sampling Methods for Topic Models [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=lda` (R package version 1.4.2)

Church, K. W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, *16*(1), 22–29.

Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M. X., & Qu, H. (2010). Context preserving dynamic word cloud visualization. In *Visualization Symposium (PacificVis), 2010 IEEE Pacific* (pp. 121–128).

Darling, W. M. (2011). A theoretical and practical implementation tutorial on topic modeling and Gibbs sampling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 642–647).

Fast, A., & Elder, J. (2014). *Text Mining Versus Text Analytics: Research brief, part ii of ii.* `http://cdn2.hubspot.net/hubfs/2176909/Resources/Whitepaper_Text_Mining_vs_Text_Analytics_IIA_Research_Brief.pdf`.

Feinerer, I., & Hornik, K. (2015). Text Mining Package [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=tm` (R package version 0.6-2)

Fellows, I. (2014). wordcloud: Word Clouds [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=wordcloud` (R package version 2.5)

Flanagan, D. (2006). *JavaScript: the Definitive Guide*. "O'Reilly Media, Inc.".

Fminer web site. (2017). `http://www.fminer.com/`. (Accessed: 19-09-2017)

Forman, G. (2008). Bns feature scaling: an improved representation over tf-idf for svm text classification. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (pp. 263–270).

Gelbukh, A. (2006). Computational linguistics and intelligent text processing. In *7th International Conference, Cicling 2006.*

Grün, B., & Hornik, K. (2011). topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, *40*(13), 1–30.

Harrison, J. (2014). RSelenium: R bindings for Selenium WebDriver [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=RSelenium` (R package version 1.3.5)

Hartigan, J. A., & Hartigan, J. (1975). *Clustering Algorithms*. Wiley New York.

Heimerl, F., Lohmann, S., Lange, S., & Ertl, T. (2014). Word cloud explorer: Text analytics based on word clouds. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on* (pp. 1833–1842).

Hirschey, J. K. (2014). Symbiotic relationships: Pragmatic acceptance of data scraping. *Berkeley Tech. LJ*, *29*, 897.

Hofmann, T. (1999a). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (pp. 289–296).

Hofmann, T. (1999b). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 50–57).

Hornik, K. (2017). StanfordCoreNLP: Stanford CoreNLP Annotation [Computer software manual]. (R package version 0.1-2)

Hornik, K., Feinerer, I., Kober, M., & Buchta, C. (2012). Spherical k-means clustering. *Journal of Statistical Software*, *50*(10), 1–22.

Hruschka, H. (1986). Market definition and segmentation using fuzzy clustering methods. *International Journal of Research in Marketing*, *3*(2), 117–134.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, *31*(8), 651–666.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc.

Kwartler, T. (2017). Hidden structures: Clustering, string distance, text vectors and topic modeling. *Text Mining in Practice with R*, 129–179.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*(2-3), 259–284.

Landers, R. N., Brusso, R. C., Cavanaugh, K. J., & Collmus, A. B. (2016). A primer on theory-driven web scraping: Automatic extraction of big data from the Internet for use in psychological research. *Psychological Methods*, *21*(4), 475.

Li, T., Sindhwani, V., Ding, C., & Zhang, Y. (2009). Knowledge transformation for cross-domain sentiment classification. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 716–717).

Litvin, S. W., Goldsmith, R. E., & Pan, B. (2008). Electronic word-of-mouth in hospitality and tourism management. *Tourism Management*, *29*(3), 458–468.

Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation & Computational Linguistics*, *11*(1-2), 22–31.

Magatti, D., Calegari, S., Ciucci, D., & Stella, F. (2009). Automatic labeling of topics. In *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on* (pp. 1227–1232).

Mcauliffe, J. D., & Blei, D. M. (2008). Supervised topic models. In *Advances in Neural Information Processing Systems* (pp. 121–128).

McLachlan, G., & Krishnan, T. (2007). *The EM algorithm and extensions* (Vol. 382). John Wiley & Sons.

Mei, Q., Shen, X., & Zhai, C. (2007). Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 490–499).

Meyer, D., Hornik, K., & Feinerer, I. (2008). Text mining infrastructure in R. *Journal of statistical software*, *25*(5), 1–54.

Miner, G. (2012). *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Academic Press.

Munzert, S., Rubba, C., Meißner, P., & Nyhuis, D. (2014). *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*. John Wiley & Sons.

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, *26*(4), 354–359.

Murtagh, F., & Legendre, P. (2014). Wards hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *Journal of Classification*, *31*(3), 274–295.

Outwit web site. (2017). `http://www.outwit.com/`. (Accessed: 19-09-2017)

Parsehub web site. (2017). `http://www.parsehub.com/`. (Accessed: 19-09-2017)

Phantomjs webstack. (2016). `http://http://phantomjs.org/`. (Accessed: 30-04-2016)

PhoCusWright: "custom survey research engagement", prepared for tripadvisor. (2014). `http://www.un-industria.it/Public/Doc/tripadvisor_unindustria_roma-19032014.pdf`. (Delivered: 19-03-2014)

Plisson, J., Lavrac, N., & Mladenić, D. (2004). A rule based approach to word lemmatization. In *Proceedings of IS04.*

Poggi, N., Berral, J. L., Moreno, T., Gavalda, R., & Torres, J. (2007). Automatic detection and banning of content stealing bots for e-commerce. In *NIPS 2007 Workshop on Machine Learning in Adversarial Environments for Computer Security* (Vol. 2).

Popescu, I.-I., Mačutek, J., & Altmann, G. (2009). *Aspects of Word Frequencies*. RAM-Verlag Lüdenscheid.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130–137.

R Core Team. (2017). R: A Language and Environment for Statistical Computing [Computer software manual]. Vienna, Austria. Retrieved from `http://www.R-project.org/`

Ramos, J. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first Instructional Conference on Machine Learning* (Vol. 242, pp. 133–142).

Robinson, S. R., Robinson, T., & Burson, R. (2010, May 28). *Trained predictive services to interdict undesired website accesses.* Google Patents. (US Patent App. 12/789,493)

Rokach, L., & Maimon, O. (2005). Clustering methods. In *Data Mining and Knowledge Discovery Handbook* (pp. 321–352). Springer.

Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Scanfeld, D., Scanfeld, V., & Larson, E. L. (2010). Dissemination of health information through social networks: Twitter and antibiotics. *American Journal of Infection Control*, *38*(3), 182–188.

Selenium web tool. (2017). http://www.seleniumhq.org/. (Accessed: 30-04-2016)

Struhl, S. (2015). *Practical Text Analytics: Interpreting Text and Unstructured Data for Business Intelligence*. Kogan Page Publishers.

Tewarson, R. P. (1973). *Sparse Matrices*. Academic Press New York.

Toman, M., Tesar, R., & Jezek, K. (2006). Influence of word normalization on text classification. *Proceedings of International Conference on Multidisciplinary Information Sciences and Technologies*, *4*, 354–358.

Tripadvisor web site. (2017). http://www.tripadvisor.com/. (Accessed: 01-03-2017)

Wallach, H. M., Mimno, D. M., & McCallum, A. (2009). Rethinking LDA: Why priors matter. In *Advances in Neural Information Processing Systems* (pp. 1973–1981).

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*(58), 236-244.

Wickham, H. (2015). rvest: Easily Harvest (Scrape) Web Pages [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=rvest (R package version 0.3.1)

Xu, P., Wu, Y., Wei, E., Peng, T.-Q., Liu, S., Zhu, J. J., & Qu, H. (2013). Visual analysis of topic competition on social media. *IEEE Transactions on Visualization and Computer Graphics*, *19*(12), 2012–2021.

Yang, Y., Wilson, L., & Wang, J. (2010). Development of an automated climatic data scraping, filtering and display system. *Computers and Electronics in Agriculture*, *71*(1), 77–87.

Zhao, Y., & Karypis, G. (2002). Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management* (pp. 515–524).

Zhong, S. (2005). Efficient online spherical k-means clustering. In *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference* (Vol. 5, pp. 3180–3185).

Latent Aspect Rating Analysis: a Model-Based Approach

Abstract

This article aims to improve some existing methods of inference applied to online user-generated content of products or services containing an overall rating. The objective is to discover latent aspects and to estimate their ratings and weights. The method proposed here considers the weaknesses of the current methods and offers an alternative approach. Our model considers the Latent Dirichlet Allocation and the Latent Rating Regression, the former being applied in a sentence-based manner and the latter with a more solid statistical ground. The methodology is applied to user-generated reviews of some major hotels in Boston, Massachusetts, to illustrate the proposed procedure.

Key words: Latent rating analysis, Logistic normal distribution, Opinion analysis, Sentence-based LDA

1 Introduction

The abundance of opinionated reviews on the Internet regarding products or services requires text mining techniques, computer processing and human analysis. There are several review websites specializing in different kinds of opinions but for travel-related reviews, the most important sources are TripAdvisor, Booking, Yelp, Agoda, and so on.

A review on a touristic structure is an evaluation text emphasizing positive or negative experiences under various aspects. These aspects tend to be defined by common characteristics for all the structures of the same type. For instance, for a hotel, there can be aspects such as *Service*, *Location* or *Room*. Quantifying the whole experience in a review is usually accomplished by assigning a value to the review and this represents an overall rating. This value may be expressed on a scale from one to five where "1" represents a poor experience, and "5" an excellent one. Each aspect is emphasized to a certain degree by the weight the reviewer puts on the words she chooses to describe them. When we deal with a set of reviews with overall ratings, three characteristics may be of interest, the first one is the *major aspects* commented, another one is the *ratings on each aspect* and the last one is the *relative weights placed on different aspects* by the reviewer (Wang et al., 2010). Indeed, we could think of a generative model to idealize the way a review is created. When a person prepares a review, firstly she decides on the aspects and

consequently chooses semantically meaningful words to describe them. Secondly, based on the sentiment of these words, she gives a rating for each aspect. Eventually, an overall rating is assigned as a weighted sum of all the composing aspect ratings (Wang et al., 2010). The weight indicates the emphasis the reviewer has associated to each aspect. Typically, for each review, the aspects are determined *a priori* but their ratings and the weights are latent.

In this article we propose a two-step inferential method. The first step infers the aspects with the application of LDA considering each sentence as a document. The second step estimates the ratings and the weights for the discovered aspects through a modified Latent Rating Regression (LRR) model.

This paper is composed by six sections. In the next session we review the existing algorithms and methods, namely the LDA (Blei et al., 2003) topic model, LARA (Wang et al., 2010) and LARAM (Wang et al., 2011). In the third section we present our model proposal. In the fourth section we detail our model explaining the parameter estimation for document-specific parameters and corpus-level parameters. The fifth section is an application of the previously described methodology for illustration purposes. The last section is dedicated to the conclusions.

2 Review of existing approaches

Often, an online review of a service or product is accompanied by an overall judgment expressed by a rating. Sometimes, the rating is missing and in literature there are various approaches to predict it, either in the form of polarity (Cui et al., 2006; Dave et al., 2003; Devitt and Ahmad, 2007), or on a rating scale (Goldberg and Zhu, 2006; Pang and Lee, 2005). A review might contain multiple latent aspects (McAuley and Leskovec, 2013), sometimes called themes or topics, and all these aspects have their own rating. These are important issues, yet in the statistical or machine learning literature relevant for this research topic, they have been seldom considered. For the prediction of the aspect ratings, Snyder and Barzilay (2007) use their algorithm called *good grief* to model the dependencies among aspects. Titov and McDonald (2008) consider the aspect ratings to be provided in the training data. For the description of the aspects they use topics, then employ a regression model. Wang et al. (2010) assume a different scenario in which the aspect ratings are latent, as in Lu et al. (2009) who have as a goal to generate an aggregated summary with aspect ratings inferred from overall ratings.

To our knowledge, there are two latent regression models in the literature, Latent Aspect Rating Analysis (LARA) (Wang et al., 2010) and Latent Aspect Rating Analysis Model (LARAM) (Wang et al., 2011) for estimating the individual ratings on each aspect

and the weight a reviewer puts on them in opinionated reviews. Both models are proposed by the same authors, so they share some common features. The approaches discussed in the two mentioned articles have the same goal, albeit methodologically implemented in a different way. For a data set of reviews which has an overall rating, the algorithms discover the aspects, then, on each of them, analyze the opinions identifying the latent aspect ratings as well as the weight (implicitly) placed on the aspects by the user. Before going into the details of these two methods, we will briefly review the important methodology of Latent Dirichlet Allocation (Blei et al., 2003) and its variant given by Sentence-Constrained Latent Dirichlet Allocation (Büschken and Allenby, 2016). They are relevant for the LARAM approach and for the proposal of this paper, respectively.

The treatment of the various methods requires a common notation, given as follows. We start from a corpus of D documents (reviews), each with n_d words w_{dn}, $d = 1, \ldots, D$ and $n = 1, \ldots, N_d$, taken from a common vocabulary with V words. Indeed, it is straightforward to map the vocabulary into the set of indexes $\{1, \ldots, V\}$, so that $w_{dn} \in \{1, \ldots, V\}$. Each document has its own rating, r_d, typically a positive number, often of discrete nature.

Note: in the description that follows, a unified notation has been adopted, trying to unify the different notations used by the various authors; this implies that the symbols used for some of the model quantities introduced may differ from the symbols used in the original sources.

2.1 Topic models

Topic models are statistical models that help to discover hidden topics which are represented by a set of words across collections of documents. They are also used to annotate documents according to these topics and to use the annotations in order to search, organize or summarize texts (Blei and Lafferty, 2009; Blei et al., 2003; Hornik and Grün, 2011). One important topic model algorithm is Latent Dirichlet Allocation (LDA), described below.

2.1.1 Latent Dirichlet allocation

When we have a collection of text corpora and we want to discover topics in documents, the most appropriate generative probabilistic model that comes into play is LDA (Blei et al., 2003; Griffiths et al., 2005). In the LDA model, we consider a corpus made up of several documents. Each document is composed of a mixture of topics and each topic is composed of a number of words. Each word in the document, from the first to the last, belongs to one of the document's topics. The model assumptions are as follows.

For each document $d = 1, \ldots, D$, and for a fixed number K of topics

1. Draw a topic distribution, $\boldsymbol{\theta}_d \sim \text{Dir}(\gamma)$, where γ is the scalar parameter for the K-dimensional symmetric Dirichlet distribution, for which all the parameters have the same value γ.

2. For each word in the document w_{dn}, $n = 1, \ldots, N_d$

   i) Draw a topic $z_{dn} \sim \text{Multinomial}(\boldsymbol{\theta}_d)$

   ii) Draw a word from the multinomial distribution corresponding to the chosen topic, $w_{dn} \sim \text{Multinomial}(\boldsymbol{\beta}_{z_{dn}})$

The corpus-level parameter $\boldsymbol{\beta}$ is then given by a $K \times V$ matrix, each row corresponding to a different multinomial distribution on the entire vocabulary. In a Bayesian approach, a prior distribution depending on the hyperparameter λ may be assumed for $\boldsymbol{\beta}$.

Here we leave out of the model the choice of the number of words for the document N_d. The feasible solution we adopted is to assume it has a distribution with parameter totally separated from the model parameters, hence it is not essential to include such distribution in the description. Aside from this, the most striking feature of LDA is given by the so-called *bag-of-words* assumption: each word may belong to a different topic, independently of the remaining words of the documents.

The model is used to discover the different topics that the documents is composed of and the topic composition of the document. The document-topic and the topic-word distribution coefficients are usually estimated following a Bayesian approach, declined in several alternative variants. The paper that originally proposed LDA, Blei et al. (2003), introduced the Variational Approach for estimating the parameters. The popular Variational Expectaction-Maximization (VEM) algorithm (e.g. MacKay, 2003) is a commonly-used method to estimate the model parameters (Hornik and Grün, 2011; Ormerod and Wand, 2010). It might be argued that such an approach actually belongs to the family of Empirical Bayes methods, (Efron and Hastie, 2016; Jiang, 2007) rather than to fully Bayesian methods. This latter group of techniques includes instead another popular method for LDA, the Collapsed Gibbs Sampling (Griffiths and Steyvers, 2004). VEM is known to be fast enough but less precise, whereas the second one is slower and more demanding from the computational point of view but it often yields better results (Qiu et al., 2014). Several implementations of the LDA model with different improvements were proposed to enhance the original model (Blei and Lafferty, 2006; Lafferty and Blei, 2006; Wang et al., 2012).

An important enhancement of LDA is given by Supervised LDA (sLDA), introduced by Mcauliffe and Blei (2008). The sLDA specification indeed applies to the setting of

interest here, where each document has an associated rating $r_d \in \mathbb{R}$. After defining the latent topic frequencies for the d-th document $\overline{\mathbf{z}}_d$, a K-dimensional vector, the LDA model is supplemented by a regression model for the rating, conditional on the latent topic frequencies

$$r_d = \boldsymbol{\eta}^T \overline{\mathbf{z}}_d + \varepsilon_d \,, \qquad \varepsilon_d \sim \mathcal{N}(0, \sigma^2) \,. \tag{1}$$

As noted by Mcauliffe and Blei (2008), the specification (1) does not include an intercept term, since $\sum_k \overline{z}_{dk} = 1$. The authors also propose to extend the model for the rating to a GLM specification (McCullagh and Nelder, 1989), in order to handle categorical responses.

2.1.2 Sentence-constrained LDA

It is assumed that, when people generate reviews, they rarely change topic within a sentence but they do change topic across sentences, so that the words inside a sentence belong to the same topic. Büschken and Allenby (2016) demonstrated the benefits of a Sentence-Constrained LDA (SC-LDA). As argued by the authors, this alleviates to a considerable extent the bag-of-words assumption, thus resulting in a better generative model for the text composition. A related approach is studied in Balikas et al. (2016), also focusing on a sentence-based approach. The authors develop a notable summary of earlier work of the same kind, such as Chen et al. (2016). Balikas et al. (2016) propose the so-called Sentence LDA (senLDA), which is essentially equivalent to SC-LDA. The empirical results of Balikas et al. (2016) point to the superiority of their approach with respect to standard LDA. The senLDA algorithm will be better understood when SC-LDA will be introduced.

A graphical representation of the SC-LDA model is given in Figure 1. In addition to the classic LDA model, a new plate is added to emphasize the distinction between the words inside sentences from the sentences inside reviews. In addition to the previously LDA notation, a different indexing is used to keep track or the words (n) contained within the sentences (s) within each review (d), w_{dsn}. The latent variable z_{ds} is assumed to be the same for all words within the sentence and is displayed outside the word plate. It is assumed that the number of sentences in a document S_d and the number of words per sentence N_{ds} are determined independently from the topic probabilities (θ_d). The probability of topic assignment changes because all words within a sentence are used to draw the latent topic assignment, z_{ds}.

For the setting of interest here, where each document has an associated rating, it seems worth mentioning the SC-LDA Rating method, also introduced in Büschken and Allenby (2016). It is a direct extension of Supervised LDA for incorporating the rating

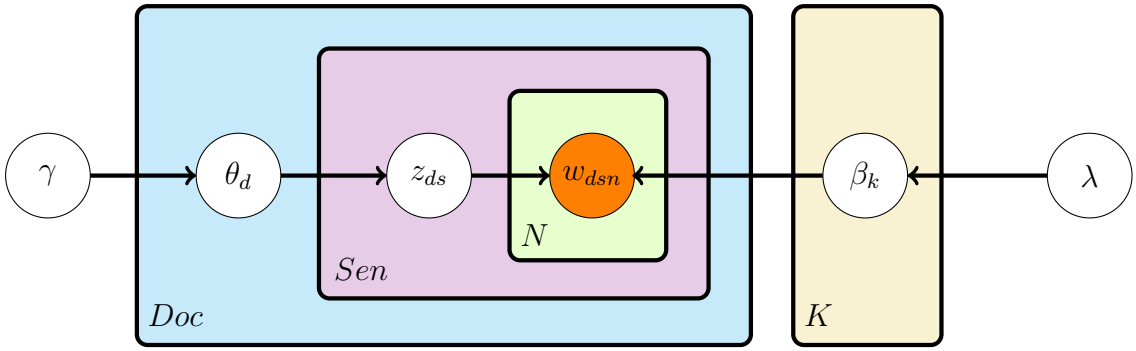

Figure 1: Graphical representation of the SC-LDA model, inspired by Büschken and Allenby (2016). Note that Sen refers to the s-th sentence within each d-th document.

prediction within Sentence-Constrained LDA. In particular, the Büschken and Allenby (2016) work adopts an ordered probit model for categorical responses (Tutz, 2011), and estimate the model parameters following a Bayesian approach. Using data from three empirical analyses, Büschken and Allenby (2016) show how SC-LDA Rating systematically outperformed sLDA in terms of out-of-sample predictions, an unsurprising result given the superiority of the sentence-based variant as a generative model.

Finally, an alternative approach to SC-LDA is given by applying LDA to each sentence, still assuming that there is just a single topic for each document. This has been proposed with some success for the analysis of microblog data, such as those of the popular Twitter social media, by the Twitter-LDA method (Zhao et al., 2011). While such an approach does not exploit the information given by the document each sentence belongs to, on the other hand it is rather practical, and it can be usefully applied by means of some twists to standard software for LDA.

2.2 LARA

The LARA method, introduced in Wang et al. (2010), consists of a two step-approach. The first step deals with the aspect discovery and it is done by an aspect segmentation procedure, dividing the reviews into aspect chunks with words that belong to a certain aspect. In particular, the output of the procedure consists in the creation of a word frequency matrix. The second step deals with the actual latent regression for finding the aspect ratings and weights using the frequency matrix built up in the first step. Let us describe the two steps with more details.

2.2.1 Aspect segmentation

The aim of the aspect segmentation is to partition the sentences of the reviews into subsets corresponding to the K aspects initially given. It was assumed that only a few

keywords are specified to describe each aspect. In order to retrieve more related words for each of them, a dedicated algorithm called Aspect Segmentation was designed (Wang et al., 2010).

The Aspect Segmentation Algorithm works as follows: given the review text, for each aspect, a few *seed words* are provided. Then, each sentence is mapped to the correspondent aspect that shares the maximum term overlapping. Based on this correspondence, dependencies are calculated between words and aspects through suitable Chi-Square statistics, attaching the words with high dependencies, one after another, to the corresponding list containing the aspect keywords. The previous steps are repeated until the aspect keyword list is unchanged or the number of iterations exceeds a certain given number (Wang et al., 2010).

The aspect segmentation yields K partitions of each review $d = 1, \ldots, D$ and represents them as a feature matrix $\mathbf{W}_d$ of size $K \times V$, where the (k, v) entry w_{dkv} is the frequency of word v ($v = 1, \ldots, V$) assigned to aspect k, normalized by the sum of words in the corresponding aspect. Feature matrices result to be extremely sparse.

2.2.2 Latent rating regression

The Latent Rating Regression model, as stated by its authors, "is a regression model that formally captures the generation process" (Wang et al., 2010). It uses a word frequency matrix $\mathbf{W}_d$ with normalized words frequency for every aspect in each review d, the outcome of the aspect segmentation phase. In the LRR model, the $\mathbf{W}_d$ are considered explanatory variables while the overall rating r_d of each review is considered the response variable. Wang et al. (2010) try to model both the latent ratings and weights of the aspects. In the LRR model the overall rating is based on the ratings on the latent aspects, which in turn are determined by the word frequencies.

The two key quantities of LRR for each document are given by the aspect ratings $\mathbf{s}_d$ and the aspect weights $\boldsymbol{\alpha}_d$, both K-dimensional vectors, which are used to predict the overall rating. Namely, the key assumption is that

$$r_d = \mathbf{s}_d^\top \boldsymbol{\alpha}_d + \varepsilon_d \,, \qquad \varepsilon_d \sim \mathcal{N}(0, \delta^2) \;, \tag{2}$$

with independence assumed across reviews.

The aspect ratings $\mathbf{s}_d$ are obtained by combining the frequency weights of the words used in each aspect with the sentiment polarity of such words. For example, if in a given review, for the *service* aspect a high frequency of the word "fantastic" is found it will probably contribute to a positive aspect rate, since the sentiment of this word for that aspect is likely to be positive. On the other hand, words like "poor", if frequent, are

likely to decrease the weight of the same aspect. To form the aspect rating, a weighted combination of the word frequencies is used. Namely, the aspect rating of document d for the k-th aspect is

$$s_{dk} = \sum_{v=1}^{V} \beta_{kv}\, w_{dkv}\,, \tag{3}$$

where β_{kv} is the (k, v) element of the matrix $\boldsymbol{\beta}$, of size $K \times V$, collecting the sentiment polarity of each word of the dictionary for each of the K aspects. Note that the possibility to have the same words but with different sentiment values for different aspects is real. Another important fact is that the matrix $\boldsymbol{\beta}$ will be sparse in most applications, since just a subset of the entire vocabulary will enter in each aspect.

The aspect weights $\boldsymbol{\alpha}_d$ are introduced to account for the heterogeneity existing between different reviews, and they are employed to form the expected value of the overall rating in equation (2), so that it is possible to interpret them as the weights given to the different aspects in forming the overall rating for the document d. Hence the various components of $\boldsymbol{\alpha}_d$ sum to 1, namely $\sum_k \alpha_{dk} = 1$, and in that respect they resemble the weights $\bar{\mathbf{z}}_d$ in (1), though their definition is different. This last property explains why model (2) does not include an intercept.

The document-specific aspect weight vectors are assumed to be normal random effects, and Wang et al. (2010) assume that

$$\boldsymbol{\alpha}_d \sim \mathcal{N}(\boldsymbol{\mu}_\alpha, \boldsymbol{\Sigma}_\alpha)\,, \tag{4}$$

with the constraints

$$0 \leq \alpha_{dk} \leq 1\,, \qquad \sum_k \alpha_{dk} = 1\,. \tag{5}$$

Putting everything together, the LRR model is composed of a set of corpus-level parameter, namely

$$\boldsymbol{\theta} = (\boldsymbol{\mu}_\alpha, \boldsymbol{\Sigma}_\alpha, \boldsymbol{\beta}, \delta^2)\,,$$

plus the document-specific parameters $\boldsymbol{\alpha}_d$, $d = 1 \ldots, D$. In Wang et al. (2010) all the parameters are estimated by maximum likelihood, after obtaining the log-likelihood function corresponding to the various assumptions made. The likelihood maximization is carried out by means of an algorithm that maximizes in turn the corpus-level parameters and the document-level parameters, respectively.

The plate notation for the LRR model is presented in Figure 2.

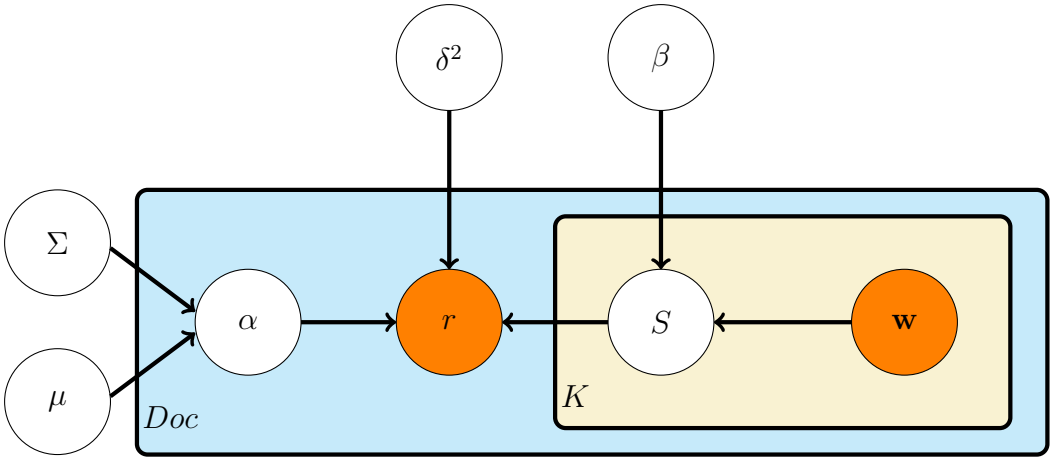

Figure 2: Graphical representation of the LRR model. Inspired by Wang et al. (2010).

2.3 LARAM

The LARAM (Latent Aspect Rating Analysis Model) approach was introduced by Wang et al. (2011). The main idea of LARAM is to model in a unified frame the discovery of the aspects, their ratings and the related weights. Thus, it proposes a model for the generation of the text and the overall rating associated to that text. Empirical evidence shows that latent topical aspects (e.g. "location" and "value for the money") are common to all the reviews albeit not all being present, the choice depending on the aspect the reviewer wanted to emphasize on. The generative model tries to capture the link between the review content and the overall rating, in a way very similar to what was done in the LDA approach. Actually, it could be argued that LARAM combines standard LDA and LARA into one model.

More in details, the generative model can be described as follows: in order to generate a review d, the user would think first of the aspects to be commented on. Each aspect would be described by words with sentiment polarities, and the review opinion on each aspect will be summarized by an aspect rating. Eventually, an overall rating r_d would be assigned to the whole review as a weighted sum of all the aspect ratings present in the review. The aspect weights $\boldsymbol{\alpha}_d$ represent the emphasis the writer puts on each aspect in the description. The model described is build up by the combination of two ideas, one being the topics discovery, the other the rating analysis used to infer the aspect ratings and weights of the discovered aspect segments in a review. Mathematically, the assumptions underlying the aspect model are as follows. For each document $d = 1, \ldots, D$, and for fixed number K of topics:

1. Draw a topic distribution, $\boldsymbol{\theta}_d \sim \text{Dir}(\gamma)$, where γ is the scalar parameter for the K-dimensional symmetric Dirichlet distribution.

2. For each word in the document w_{dn}, $n = 1, \ldots, N_d$

   i) Draw a topic $z_{dn} \sim \text{Multinomial}(\boldsymbol{\theta}_d)$

   ii) Draw a word from the multinomial distribution corresponding to the chosen topic, $w_{dn} \sim \text{Multinomial}(\boldsymbol{\varepsilon}_{z_{dn}})$.

It is straightforward to recognize that the model is exactly the same of §2.1.1, save for a change in the symbol used for the topic distribution, with $\boldsymbol{\varepsilon}$ replacing $\boldsymbol{\beta}$ since we reserve the latter symbol for the latent regression part.

In the rating analysis step, each aspect rating s_k, $k = 1, \ldots, K$, is assumed to be calculated by the aggregated sentiment over the text portions belonging to that aspect, so that the k-th rating for the d-th document is

$$s_{dk} = \sum_{v=1}^{V} \beta_{kv} \sum_{n=1}^{N_d} \Delta[w_{dn} = v, z_{dn} = k] \,. \tag{6}$$

In the previous equation, $\beta_{kv} \in \mathfrak{R}$ represents the v^{th} word's estimated sentiment polarity on aspect k, $\Delta[w_n = v, z_n = k]$ is an indicator function representing the n^{th} word in review d, which is the v^{th} entry in vocabulary V, that word being part of aspect k. Note the similarity between equations (6) and (3) in §2.2.2, which is even more apparent if we set

$$w_{dkv} = \sum_{n=1}^{N_d} \Delta[w_{dn} = v, z_{dn} = k] \,,$$

acknowledging that the latent aspects of each word z_{dn} is used here to replace the word frequencies defined by aspect segmentation in LARA.

The remaining part of the rating analysis step is exactly equal to the LRR part of LARA. Namely, the document-specific vector of aspect weights $\boldsymbol{\alpha}_d$ is introduced, model (2) is assumed for the overall rating, and the assumptions (4) and (5) are made on the aspect weights.

The fundamental peculiarity of the model is the bridge between the aspect discovery and the latent regression. More precisely, the connection between the review content $\mathbf{W}_d$ and the latent aspect ratings $\mathbf{s}_d$ is given by the set of aspect assignments $\{z_{d1}, \ldots, z_{dN_d}\}$, as made explicit in formula (6). The latent aspect assignments are introduced to associate a word with its corresponding aspect, thus the rating regression model can exploit this association to infer the latent aspect ratings and weights. Figure 3 shows a graphical representation of the model, where it is emphasized how the aspect modeling part is connected to the latent analysis model by means of the latent assignments.

As stressed in Wang et al. (2011), it should be noted that:

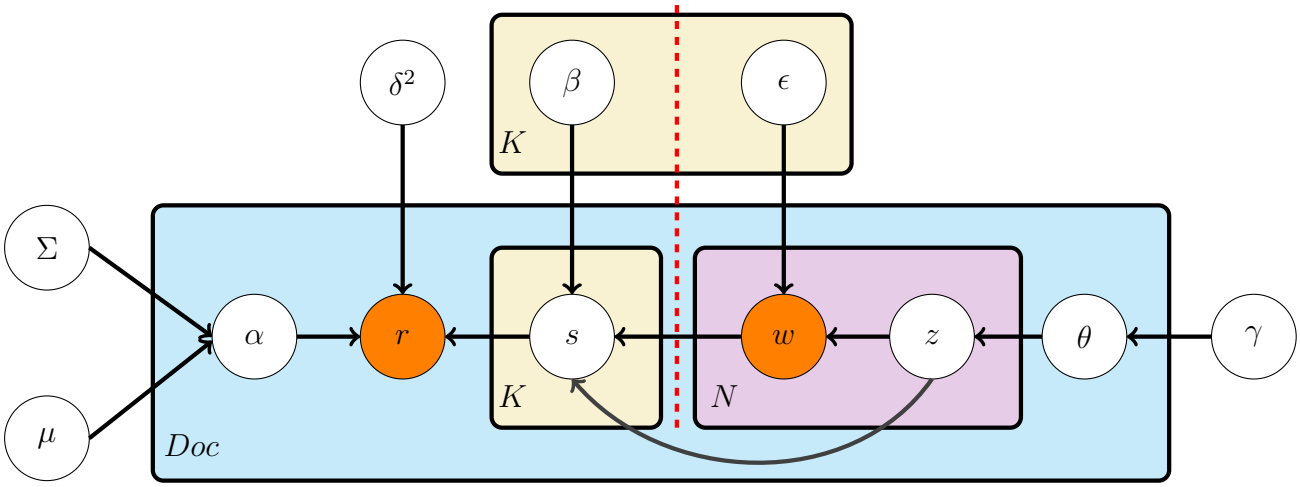

Figure 3: Graphical representation of the LARAM model. The vertical line separates the LDA-part from the LRR-part of the model. Inspired by Wang et al. (2011).

(i) The latent aspect rating is a sum over a set of random variables because each word's aspect assignment z_{dn} is random, whereas in LRR of LARA the aspect ratings are non-random;

(ii) Each word is assigned to an aspect for which the word's sentiment orientation is the most consistent with the other words belonging to the same segment, whereas the LRR model has a fixed segmentation after the discovery of the aspect keywords.

To wind up, the integrated model of LARAM comprises a set of document-specific parameters $\boldsymbol{\alpha}_d$, a set of word-specific topic assignments and a set of corpus-level parameters $\boldsymbol{\theta} = (\boldsymbol{\varepsilon}, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \delta^2)$. From the inferential point of view, the original LARAM article proposes the variational Bayes method, along the lines of Blei et al. (2003). As customary for variational methods, the corpus parameters are estimated using a VEM algorithm, ending up with posterior estimates of latent aspect assignments for each word and aspect weights as well.

2.4 Comparative analysis of existing methods

The methods illustrated above have strengths and weaknesses. The two LDA-based methods, sLDA and SC-LDA Rating, have the advantage of being quite simple, as they can be seen as just an enhancement of their unsupervised counterparts. On the other hand, model (1) has little flexibility, since every aspect frequency is associated to a single fixed coefficient η_k, ending up with a rating prediction which only depends on the proportion of words of each aspect identified in a given document. This feature is shared by SC-LDA Rating, which however has the important feature of being able to overcome

the bag-of-words assumption, an important issue to be kept in mind. The fact that either SC-LDA Rating or sLDA may be cast in GLM form for handling ordinal responses is a positive feature, whose importance, however, should not be overstated. Indeed, even in those cases where the overall rating r_d is discrete (1-5 stars, say), the assumption of equally-spaced categories is in most cases quite sensible, hence treating it as normally distributed amounts to a reasonable approximation.

The evaluation of LARA and LARAM is more delicate, since they are more sophisticated methods. The main striking point of both methods compared to sLDA and SC-LDA Rating is the adoption of document-specific random effects, which correspond to aspect weights and carry on all the advantages of random effects modelling (Efron and Hastie, 2016; Jiang, 2007; Tutz, 2011). Another positive feature of these two methods is that they provide the document-specific aspect ratings as further outcome.

LARA has the advantage over LARAM of being scalable. The two-step approach gives the method a good deal of flexibility. On the negative side, the aspect segmentation algorithm has strong limitations, being based on observed associations which are often weak, as is typical of sparse settings.

LARAM improves on LARA by replacing Aspect Segmentation with a generative model which actually coincides with the well-established LDA model. The fact that the two phases of aspect allocation and latent rating regression are done simultaneously makes things more complicated, and indeed parameter estimation in LARAM may become rather challenging. At the same time, the integrated nature of the model has the advantage of correctly taking the uncertainty of the first phase into account at the second phase. The fact that the latent aspect association is (partially) driven by the overall rating, which is implicit in the procedure, appears to us as a possible rigidity of the procedure.

Both LARA and LARAM are based on the bag-of-words assumption, the latter in a more explicit form. Moreover, both methods make use of the assumptions (4) and (5) which should be taken, at best, as an approximation. Although they may still give good results in practice, we will see how the theory of probability distributions on the simplex provides more satisfactory, and safer, mathematical models.

3 Main proposal

Our proposal consists of sentence-based topic modelling followed by *bona fide* random-effects rating regression. Indeed, based on the considerations illustrated in the previous section, the idea is to combine the best aspects of the existing methods. The aim is to retain the flexibility and the agility of the two-step approach of LARA, which is the

method closest to our proposal. In fact, despite the integrated approach of LARAM might have some good properties in principle, we believe that in practice the overall task aimed at is very complex, and a *divide-et-impera* strategy is highly recommendable. The two steps are as follows:

Step i.

We apply a SC-LDA to the entire corpus, ending up with the assignment of each sentence to a certain aspect out of given number K. This allows us to form a feature matrix $\mathbf{W}_d$ for each document, where each word of the dictionary gets an assigned value for each aspect. Such value could be an empirical frequency, or just an indicator function flagging the appearance of that word in the document for a given aspect. We can then proceed to define aspect rating following either (3) or (6), respectively. In the latter case,

$$s_{dk} = \sum_{v=1}^{V} \beta_{kv} \sum_{n=1}^{N_d} \Delta[w_{dn} = v, \widehat{z}_{dn} = k] = \sum_{v=1}^{V} \beta_{kv}\, \widehat{w}_{dkv}\,, \tag{7}$$

where $\widehat{z}_{dn}$ is the aspect assigned to word n of document d after fitting the SC-LDA model, and the notation $\widehat{w}_{dkv} = \sum_{n=1}^{N_d} \Delta[w_{dn} = v, \widehat{z}_{dn} = k]$ is used to stress the dependence on the estimated topic model. Should the specification (3) be preferred, its usage just requires some simple changes to the definition of $\widehat{w}_{dkv}$.

Two possible twists may be worth considering. The first one is to replace SC-LDA in Figure 1 by the simplest LDA applied to sentences, along the lines of Twitter-LDA, as mentioned in §2.1.2. The second possibility is to supplement the topic choice model with some *seed words* (Jagarlamudi et al., 2012; Ramesh et al., 2014), which may be simple to elicit for the analysis of hotel reviews or other tourism-related reviews. The fact that a sentence-based topic model is adopted is very important, since this corresponds to a reasonable model for the sentence creation that overcomes the bag-of-words assumption.

Step ii.

The second part of the proposal follows the lines of the LRR step of LARA/LARAM, but with some crucial differences, that can have a major impact on the performances of the methodology. We keep model (2) for the overall rating

$$r_d = \mathbf{s}_d^\top \boldsymbol{\alpha}_d + \varepsilon_d\,, \qquad \varepsilon_d \sim \mathcal{N}(0, \delta^2)\ , \tag{8}$$

with independence assumed across different documents, but we make different assumptions for the aspect weights $\boldsymbol{\alpha}_d$. We still treat them as random effects, retaining all the desirable implications of this choice, but we replace the assumption (4) with a mathematically correct assumption derived from the theory of compositional data, for which the monograph by Aitchison (1986) provides a comprehensive treatment. Namely, we as-

sume that $\boldsymbol{\alpha}_d$ has a logistic normal distribution on the simplex, with probability density function that, aside from some fixed constants, is given by (see Aitchison, 1986, Chapter 5)

$$p\left(\boldsymbol{\alpha}_d; \boldsymbol{\mu}_\alpha, \boldsymbol{\Sigma}_\alpha\right) \propto \left|\boldsymbol{\Sigma}_\alpha\right|^{-1/2} \left(\alpha_{d1} \cdots \alpha_{dK}\right)^{-1} \exp\left[-\frac{1}{2}\left\{\log\left(\boldsymbol{\alpha}_{d(-K)}/\alpha_{dK}\right) - \boldsymbol{\mu}_\alpha\right\}^\top \boldsymbol{\Sigma}_\alpha^{-1}\left\{\log\left(\boldsymbol{\alpha}_{d(-K)}/\alpha_{dK}\right) - \boldsymbol{\mu}_\alpha\right\}\right], \quad (9)$$

with $\sum_{k=1}^K \alpha_{dk} = 1$. Here $\boldsymbol{\alpha}_{d(-K)}$ refers to the first $(K-1)$ component of the vector $\boldsymbol{\alpha}_d$ and likewise $\boldsymbol{\mu}_\alpha$ is a $(K-1)$-dimensional vector and $\boldsymbol{\Sigma}_\alpha$ is a symmetric positive definite matrix of size $(K-1) \times (K-1)$. The book by Aitchison (1986) illustrates several contexts where model (9) is rather useful, achieving much more flexibility than the Dirichlet distribution for $\boldsymbol{\alpha}_d$, which is a natural alternative model. However, it is apparent that the density function (9) is not symmetric with respect to all the elements of $\boldsymbol{\alpha}_d$, since the last element α_{dK} is treated differently from the others. For the setting of interest here, where the random effects distribution for $\boldsymbol{\alpha}_d$ has to be compounded with model (8), this is a rather serious limitation, since the resulting inference would depend on the labelling of the aspects, which is totally unappealing.

Fortunately, the theory of statistical models for compositional data provides a straightforward way to fix this problem, by the recourse to an alternative form for (9). We first define

$$\boldsymbol{\eta}_d = \log\left\{\boldsymbol{\alpha}_d / g(\boldsymbol{\alpha}_d)\right\}, \quad (10)$$

where $g(\cdot)$ is the geometric mean of its argument, namely $g(\mathbf{x}) = (x_1 \cdots x_K)^{1/K}$, and then we re-express the model in terms of a distribution for the vector $\boldsymbol{\eta}_d$.

In particular, we assume a singular multivariate normal distribution for $\boldsymbol{\eta}_d$, with K-dimensional mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ of size $K \times K$ (see Rao, 1973, §8.a4); see also Srivastava and von Rosen (2002), namely

$$\boldsymbol{\eta}_d \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \quad (11)$$

Indeed, since

$$\sum_{k=1}^K \eta_{dk} = 0, \quad (12)$$

it also follows that $\sum_{k=1}^K \mu_k = 0$ and the matrix $\boldsymbol{\Sigma}$ has rank $(K-1)$, with $(K-1)$ positive eigenvalues and one nil eigenvalue. The inverse map of (10) is given by

$$\alpha_{dk} = \frac{\exp\{\eta_{dk}\}}{\sum_{k=1}^K \exp\{\eta_{dk}\}}, \qquad k = 1, \ldots, K, \quad (13)$$

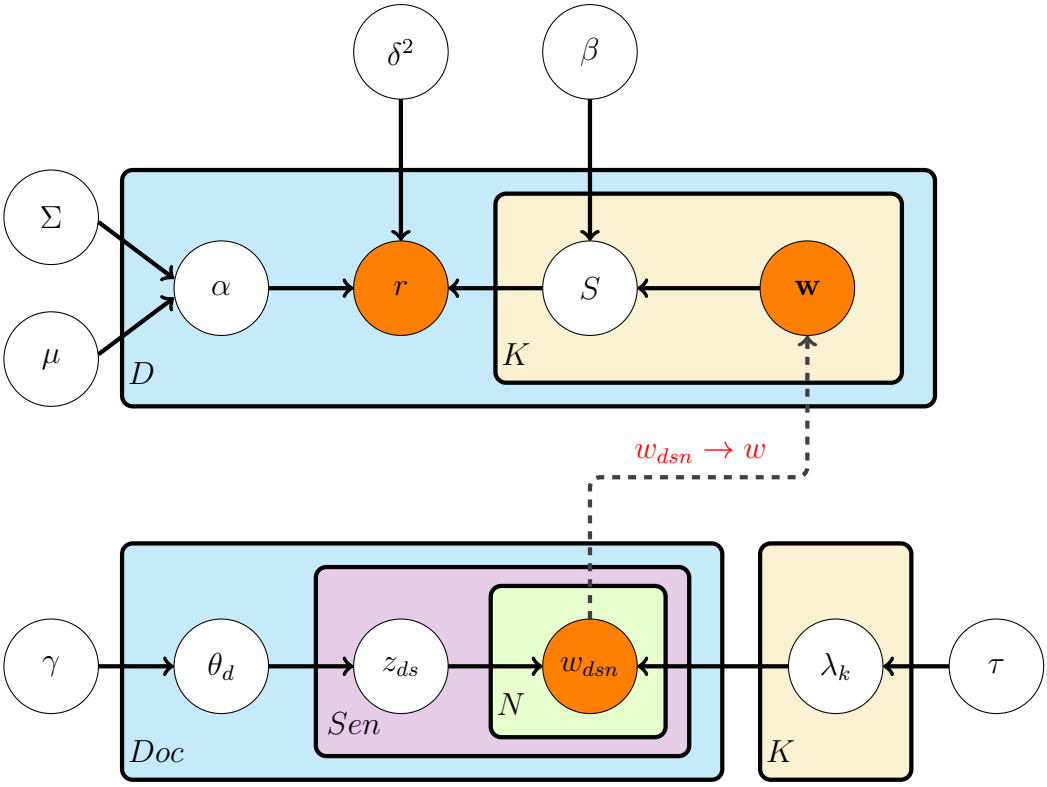

Figure 4: Graphical representation of our two-step approach. Note that *Sen* in the lower plate refers to the s-th sentence within the d-th document, whereas S in the upper plate refers to the aspect rating vectors for the d-th document.

confirming immediately that indeed $\sum_{k=1}^{K} \alpha_{dk} = 1$. We note that the singular multivariate normal distribution employed for $\boldsymbol{\eta}_d$ has found useful applications in statistics (Gelman et al., 1996); in the context of compositional data, the covariance matrix $\boldsymbol{\Sigma}$ is also known as the *centred logratio covariance matrix* (see Aitchison, 1986, Chapter 4). Using standard results of compositional data theory (see Aitchison, 1986, Chapter 5), we can re-express (9) as

$$p\left(\boldsymbol{\alpha}_d; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \propto \left| \prod_{k=1}^{(K-1)} \sigma_k \right|^{-1/2} \left(\alpha_{d1} \cdots \alpha_{dK}\right)^{-1} \exp\left[-\frac{1}{2} \left\{\boldsymbol{\eta}_d(\boldsymbol{\alpha}_d) - \boldsymbol{\mu}\right\}^{\top} \boldsymbol{\Sigma}^{-} \left\{\boldsymbol{\eta}_d(\boldsymbol{\alpha}_d) - \boldsymbol{\mu}\right\} \right], \quad (14)$$

where $\sigma_1, \ldots, \sigma_{(K-1)}$ are the positive eigenvalues of $\boldsymbol{\Sigma}$, the function $\boldsymbol{\eta}_d(\boldsymbol{\alpha}_d)$ is given by (10) and $\boldsymbol{\Sigma}^{-}$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{\Sigma}$. The density (14) is the form that is employed in the following proposal.

A summary of the entire approach is provided in Figure 4, which essentially combines together Figure (1) and Figure (2).

4 Parameter estimation

In order to apply the two-step approach proposed in the previous section, we need to consider the two steps in turn. The first step entails the estimation of a Sentence-based LDA model, a task that can be carried out in a relatively straightforward way, as alluded to in the previous section. The estimation of the LRR model of Step ii. is more involved, and will be illustrated in the remaining part of the section.

Like for LARA/LARAM, also our LRR model comprises both document-specific parameters $\boldsymbol{\alpha}_d$ and corpus-level parameters, given by

$$\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \delta^2) \,.$$

Similarly to Wang et al. (2010), we estimate both $\boldsymbol{\alpha}_d$ and $\boldsymbol{\theta}$ by maximum likelihood estimation, with a further regularization for what concerns the estimation of $\boldsymbol{\beta}$. Inferentially, this could be seen as an application of Henderson's mixed equations (Robinson, 1991), where fixed effects and random effects are jointly estimated based on observed normal data. The frequentist theory of Maximum a Posterior estimation (MPE) (see Jiang, 2007, Chapter 3) provides an alternative way to justify this method, which could also be interpreted as a special case of h-likelihood estimation (Lee and Nelder, 2009; Lee et al., 2006).

Following Jiang (2007, §3.6.1) the estimation algorithm consists in iterating two steps. Let $\boldsymbol{\alpha}$ be the vector collecting all the document-specific aspect weights, namely $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_D)$, and $\mathbf{r} = (r_1, \ldots, r_D)$ the vector collecting all the document overall ratings. Then we define $L_J(\boldsymbol{\alpha}, \boldsymbol{\theta}) = p(\mathbf{r}, \boldsymbol{\alpha}|\boldsymbol{\theta})$ the joint density function of $\mathbf{r}$ and $\boldsymbol{\alpha}$ based on (8) and (14), assuming that the feature matrices entering (7) are known after Step i. of the previous section.

The overall maximization of $L_J(\boldsymbol{\alpha}, \boldsymbol{\theta})$ can be performed by alternating between maximizing $L_J(\boldsymbol{\alpha}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\alpha}$ (for fixed $\boldsymbol{\theta}$) and maximizing $L_J(\boldsymbol{\alpha}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ after updating $\boldsymbol{\alpha}$, a well-know optimization approach also known in statistics as *Zigzag* of *full Gauss-Seidel* approach (Smyth, 1996). The two steps are then iterated until a convergence criterion is met. In what follows these two maximization problems will be illustrated in detail.

4.1 Document-specific parameters

For each document d, the estimate of the document-specific parameters $\boldsymbol{\alpha}_d$ for fixed corpus-level parameters corresponds to

$$\widehat{\boldsymbol{\alpha}}_d = \underset{\boldsymbol{\alpha}_d}{\operatorname{argmax}} - \frac{1}{2\,\delta^2}\,(r_d - \mathbf{s}_d^\top \boldsymbol{\alpha}_d)^2 + \log p(\boldsymbol{\alpha}_d; \boldsymbol{\mu}, \boldsymbol{\Sigma})\,, \qquad 0 \leq \alpha_{dk} \leq 1\,, \qquad \sum_k \alpha_{dk} = 1\,. \tag{15}$$

It is actually more convenient to carry out the maximization with respect to $\boldsymbol{\eta}_d$, as defined by (10). Namely, we solve

$$\widehat{\boldsymbol{\eta}}_d = \underset{\boldsymbol{\eta}_d}{\operatorname{argmax}} - \frac{1}{2\,\delta^2}\,\{r_d - \mathbf{s}_d^\top \boldsymbol{\alpha}_d(\boldsymbol{\eta}_d)\}^2 + \log p\{\boldsymbol{\alpha}_d(\boldsymbol{\eta}_d); \boldsymbol{\mu}, \boldsymbol{\Sigma}\}\,, \qquad \sum_k \eta_{dk} = 0\,, \tag{16}$$

where $\boldsymbol{\alpha}_d(\boldsymbol{\eta}_d)$ is given by (13). Once $\widehat{\boldsymbol{\eta}}_d$ is obtained, then we simply get $\widehat{\boldsymbol{\alpha}}_d = \boldsymbol{\alpha}_d(\widehat{\boldsymbol{\eta}}_d)$. A more precise notation would be $\widehat{\boldsymbol{\alpha}}_d(\boldsymbol{\theta})$, but for the sake of simplicity the dependence on $\boldsymbol{\theta}$ is implicit.

The optimization problem (16) is numerically simpler to solve than (15), but it should be noted that it still entails a constrained optimization, a task requiring some care (Nocedal and Wright, 2006). Such complication follows from the requirement to keep the model specification to be invariant with respect to the choice of the aspect labelling, as mentioned in the previous section in connection with (9) and (14). It is possible to verify that such invariance cannot be achieved without imposing a symmetric constraints on the components of $\boldsymbol{\alpha}_d$ or, alternatively, on those of $\boldsymbol{\eta}_d$.

4.2 Corpus-level parameters

For what concerns the corpus-level parameters $\boldsymbol{\theta}$, we have to distinguish between the parameters entering the distribution of the overall rating, $\boldsymbol{\beta}$ and δ^2, and those entering the distribution of the random aspect weights $\boldsymbol{\alpha}_d$, namely $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

For $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, it is simple to verify that, once we fix the document specific parameters, the estimation problem is equivalent to the estimation of the mean vector and the covariance matrix of a singular normal vector (11) based on a sample of size D given by $\widehat{\boldsymbol{\eta}}_1, \ldots, \widehat{\boldsymbol{\eta}}_D$. Standard theory of the multivariate normal distribution readily gives the results that follow, which hold also in the singular case (Rao, 1973, p.532),

$$\widehat{\boldsymbol{\mu}} = \frac{1}{D}\sum_{d=1}^{D} \widehat{\boldsymbol{\eta}}_d\,, \qquad\qquad \widehat{\boldsymbol{\Sigma}} = \frac{1}{D}\sum_{d=1}^{D} (\widehat{\boldsymbol{\eta}}_d - \widehat{\boldsymbol{\mu}})(\widehat{\boldsymbol{\eta}}_d - \widehat{\boldsymbol{\mu}})^\top\,.$$

The estimation of $\boldsymbol{\beta}$ requires a more cautious approach. In principle, once the aspect weights $\boldsymbol{\alpha}$ are held fixed, the model for the overall rating r_d given by (8) is just a normal

linear regression model, but we need first to identify precisely the design matrix of the model, and then to specify a suitable method to estimate the model coefficients.

For what concerns the design matrix, after letting $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$ be the V-dimensional vectors which are the rows of the matrix $\boldsymbol{\beta}$, it follows that (7) can be re-written as

$$s_{dk} = \mathbf{c}_{dk}^{\top} \boldsymbol{\beta}_k \qquad k = 1, \ldots, K \,,$$

with $\mathbf{c}_{dk}$ is V-dimensional vector given by $\mathbf{c}_{dk} = (\widehat{w}_{dk1}, \ldots, \widehat{w}_{dkV})^{\top}$. Then we can re-express the linear predictor entering (8) as

$$\mathbf{s}_d^{\top} \boldsymbol{\alpha}_d = \sum_{k=1}^{K} \alpha_{dk}\, s_{dk} = \sum_{k=1}^{K} \alpha_{dk}\, \mathbf{c}_{dk}^{\top} \boldsymbol{\beta}_k = \sum_{k=1}^{K} \mathbf{x}_{dk}^{\top} \boldsymbol{\beta}_k \,,$$

with $\mathbf{x}_{dk} = \alpha_{dk}\, \mathbf{c}_{dk}$. Finally, after stacking all the K vectors $\boldsymbol{\beta}_k$ together into the $K \times V$-dimensional vector $\boldsymbol{\beta}^{(s)}$ given by

$$\boldsymbol{\beta}^{(s)} = \left[\boldsymbol{\beta}_1^{\top}, \ldots, \boldsymbol{\beta}_K^{\top}\right]^{\top} ,$$

and, likewise, stacking all the K vectors $\mathbf{x}_{dk}$ into

$$\mathbf{x}_d^{(s)} = \left[\mathbf{x}_{d1}^{\top}, \ldots, \mathbf{x}_{dK}^{\top}\right]^{\top} ,$$

we get

$$r_d = \{\mathbf{x}_d^{(s)}\}^{\top} \boldsymbol{\beta}^{(s)} \,.$$

In matrix form, once we let $\mathbf{X}_{\alpha}^{(s)}$ be the matrix of size $D \times (K\,V)$ collecting together all the row vectors $\mathbf{x}_d^{(s)}$, $d = 1, \ldots, D$, we can re-express model (8) in a more compact form

$$\mathbf{r} = \mathbf{X}_{\boldsymbol{\alpha}}^{(s)}\, \boldsymbol{\beta}^{(s)} + \boldsymbol{\varepsilon} \,, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \delta^2\, \mathbf{I}_D) \,. \tag{17}$$

Equation (17) clearly shows that the model of interest, for fixed aspect weights, is just a linear regression model. Despite what is written in Wang et al. (2010), OLS estimation of $\boldsymbol{\beta}$ is not a doable option, since the design matrix $\mathbf{X}_{\boldsymbol{\alpha}}^{(s)}$ is extremely sparse, and the size of the coefficients vectors will be of an order comparable with the sample size D in many applications. Indeed, it should be kept in mind that the value V will be typically very large, and therefore same sort of regularization is called for. Among the methods available (Efron and Hastie, 2016; Friedman et al., 2001; Tutz, 2011), ridge regression seems the most suitable one, compared to popular alternatives such as the lasso or elastic net. Indeed, due to the nature of the coefficient matrix $\boldsymbol{\beta}$, corresponding to the sentiment polarity of words within each aspect, forcing many non-null components of $\boldsymbol{\beta}$ to

be shrunken towards zero appears to be a fully sensible decision. We note in passing that also the LARA authors (Wang et al., 2010) adopted a sort of ridge regression to regularize parameter estimation. This fact was not mentioned in the article or in Zhai and Massung (2016), but we realized it by studying the Java code made available by the authors at `http://www.cs.virginia.edu/~hw5x/Codes/LARA.zip`. Ridge regression amounts to choose $\boldsymbol{\beta}^{(s)}$ such as

$$\widehat{\boldsymbol{\beta}}_{\lambda}^{(s)} = \underset{\boldsymbol{\beta}^{(s)}}{\operatorname{argmin}} \left\{\mathbf{r} - \mathbf{X}_{\boldsymbol{\alpha}}^{(s)} \boldsymbol{\beta}^{(s)}\right\}^{\top} \left\{\mathbf{r} - \mathbf{X}_{\boldsymbol{\alpha}}^{(s)} \boldsymbol{\beta}^{(s)}\right\} + \lambda \left\|\boldsymbol{\beta}^{(s)}\right\|^2, \tag{18}$$

where λ is a tuning parameter. An empirical Bayes interpretation is also possible, by viewing (18) as derived from the assumption $\boldsymbol{\beta}^{(s)} \sim \mathcal{N}\left(\mathbf{0}, \frac{\delta^2}{\lambda} \mathbf{I}_{KV}\right)$ (Efron and Hastie, 2016, p. 98). Once $\boldsymbol{\beta}$ is estimated, an estimate of δ^2 can be obtained as the mean of the squared residuals.

Note that ridge regression would require to choose a scale for the covariates, usually obtained by some standardization of the columns of the design matrix, but here due to the special nature of $\mathbf{X}_{\boldsymbol{\alpha}}^{(s)}$ working on the original scale is also a sensible option. In case a standardization is chosen, some attention is required due to the sparsity of $\mathbf{X}_{\boldsymbol{\alpha}}^{(s)}$.

The selection of a value for the tuning parameter λ deserves some care. In principle, it should be updated at each iteration of the estimation algorithm and set to a value that ensures a good performance with respect to some chosen criterion, such as the mean squared error of prediction estimated by cross-validation, see Friedman et al. (2001, p. 37). In practice, such an approach is computationally intensive. An alternative approach is the one considering a fixed value for λ computed as the best parameter choice if all the elements of $\boldsymbol{\alpha}_d$ are set equal to their mean value $1/K$, and then keeping it constant across the estimation. Note that the choice of estimating $\boldsymbol{\beta}$ by ridge regression renders immaterial the fact that for the MPE method some bias of the corpus-level parameters may obtain when random-effects are maximized rather than integrated out; see the discussion of Lee and Nelder (1996), and Jiang (2007, p. 136).

5 Results

The target data for this analysis were the reviews of the major hotels in Boston Massachussets and they are the same data already employed for illustration purposes in the first article of this thesis. For the purpose of our analysis we have used the overall rating and the text of the reviews. The total number of the hotels was 79, each containing from 14 to 5 382 reviews, giving an overall sample size of 93 268 reviews. After the dataset file was prepared, the next step was to subdivide the reviews into sentences.

In the following subsections we present the application of our algorithm.

5.1 LDA application

Following data gathering and cleaning, we performed LDA. An empirical semantic labeling of the topics based on the top 20 most frequent words in each of them was accomplished. The number of aspects K in this case are 5, corresponding to: 1 - *Room*, 2 - *Staff*, 3 - *Experience*, 4 - *Location*, 5 - *Value*. A color code was assigned to each topic. The result can be seen in Table 1.

Room	Staff	Experience	Location	Value
bed	staff	great	walk	breakfast
comfort	help	locat	locat	dollar
clean	friend	view	great	servic
bathroom	servic	good	restaur	free
small	frontdesk	nice	close	good
nice	great	servic	street	park
shower	back	recommend	shop	restaur
well	arriv	place	minut	bar
size	concierg	staff	distanc	food
larg	make	clean	right	coffe
good	checkin	price	park	great
door	trip	citi	airport	charg
great	call	look	easi	nice
floor	made	high	station	wifi
tvset	just	floor	just	lobbi
spacious	check	area	within	price
work	visit	overal	area	morn
wall	need	beauti	conveni	book
need	nice	love	block	offer
water	alway	well	away	just

Table 1: LDA application to the corpus of the dataset.

By joining the LDA results with the rating we obtained the word frequency matrix **W**. In Table 2 an excerpt of it is reported, displaying the last portion of review 2 and the initial part of review 3. The topic column displays the topic number estimated by LDA for each sentence, and its background color flags the corresponding aspect.

The table is composed of six columns:

- `id`: a simple sequential id number starting from the first word in the first review ending with the last word in the last review;
- `doc`: represents the review number;

id	doc	sent	index	topic	count	rating
⋮	⋮	⋮	⋮	⋮	⋮	⋮
117	2	7	13964	4	1	5
118	2	7	15034	4	1	5
119	2	7	16486	4	1	5
120	2	7	17113	4	1	5
121	3	1	9780	3	1	3
122	3	1	7338	3	1	3
123	3	2	7338	3	1	3
124	3	2	14718	3	1	3
125	3	4	7338	3	1	3
126	3	4	1486	3	1	3
127	3	5	202	5	1	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 2: Word frequency matrix.

- `sent`: represents the sentence number within the review;
- `index`: represents the position of the word in the corpus vocabulary;
- `topic`: shows the topic to which the word in that specific review belongs to;
- `count`: counts how many times that specific word is present inside the corresponding document and topic;
- `rating`: shows the overall rating of the document.

The words and their position inside a vocabulary can be observed in Table 3, which reports two variables, `index` and `word`, comprising all the words inside the whole corpus, taken only once and stored in alphabetic order.

`index`	...	2606	2607	2608	2609	2610	2611	2612	2613	...
`word`	...	care	career	careful	careless	caretalk	careworn	carey	cargo	...

Table 3: Word vector.

After the completion of the first step of the algorithm, we proceed to the estimation of the latent rating regression model.

5.2 Model estimation

In this subsection we use the results previously obtained from the LDA application, and then proceed to the estimation of the model parameters.

5.2.1 Corpus-level parameters

As seen in the previous sections, the corpus level parameters are $\boldsymbol{\beta}$, δ^2, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. $\boldsymbol{\beta}$ consists of the polarities of all the words in the vocabulary over the 5 aspects, arranged in a $5 \times V$ matrix. A word can appear in all the aspects but with different polarities. The sentiment can be negative, positive, or nil, the higher the number, the more extreme the value. As an example, in Table 4 we report a portion of the estimated $\widehat{\boldsymbol{\beta}}$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0.00	0.00	0.00	0.00	0.00	0.00	0.40	-0.43	1.13	-0.00	0.00	0.00	0.00	0.00	0.49	0.00	0.11
2	0.00	0.00	0.45	0.21	1.59	0.41	-0.97	0.66	0.00	0.00	0.25	0.00	0.13	-0.27	0.07	0.00	0.73
3	0.00	0.00	0.00	-0.00	0.36	0.22	-0.49	-1.25	0.00	0.02	0.00	0.09	-0.00	-0.03	0.00	0.35	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.61	-0.00	0.00	0.00	0.27	0.14	0.00	0.00	0.00	1.04
5	0.28	0.00	0.00	0.00	0.00	0.57	0.04	0.11	0.00	0.02	0.00	0.00	0.88	0.00	0.00	0.00	0.30

Table 4: Estimated $\widehat{\boldsymbol{\beta}}$.

Summarizing the values for $\widehat{\boldsymbol{\beta}}$ we can observe the prevalence of negative values over the positive ones (see Table 5), thus we can say that reviews have a role of complaint rather than praise. This behavior is expected due to the reasons a person would leave a review, most of the time when they have negative experiences (Melián-González et al., 2013; Vásquez, 2011). Most $\widehat{\boldsymbol{\beta}}$ values are just left equal to zero, or to values very close to it, since not enough data are available for their estimation. This is an effect of using a regularization method such as ridge regression.

	Room	Service	Experience	Location	Value
$\widehat{\boldsymbol{\beta}} > 0$	26.04%	24.5%	21.27%	18.67%	29.15%
$\widehat{\boldsymbol{\beta}} < 0$	28.35%	34.92%	28.27%	25.39%	31.12%
$\widehat{\boldsymbol{\beta}} = 0$	45.61%	40.58%	50.46%	55.94%	39.73%

Table 5: $\widehat{\boldsymbol{\beta}}$ values, polarities.

Figure 5 visualizes the empirical distribution of the estimated $\widehat{\boldsymbol{\beta}}$ values. We notice that all the histograms have a peak around the zero value, but have otherwise a symmetric shape.

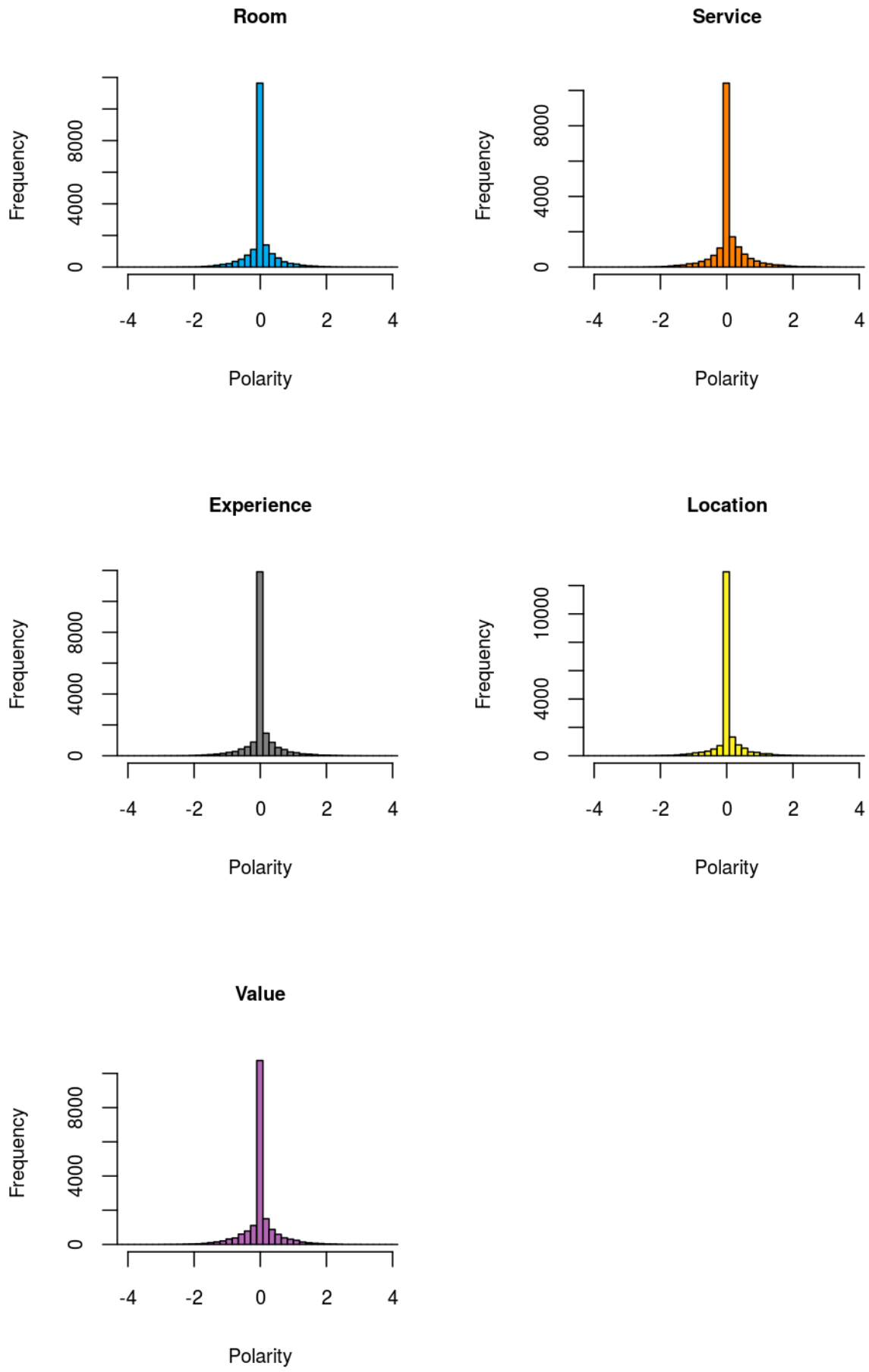

Figure 5: Histograms of $\widehat{\boldsymbol{\beta}}$ values.

Figure 6 reports the pairwise correlation for the rows of the $\widehat{\boldsymbol{\beta}}$ matrix. It is apparent there is a positive correlation in all cases, though the values are generally low due to the large portion of estimated coefficients equal to zero. This amounts to say that the estimated word polarity across different aspects tends to be coherent, though local differences for specific words may arise.

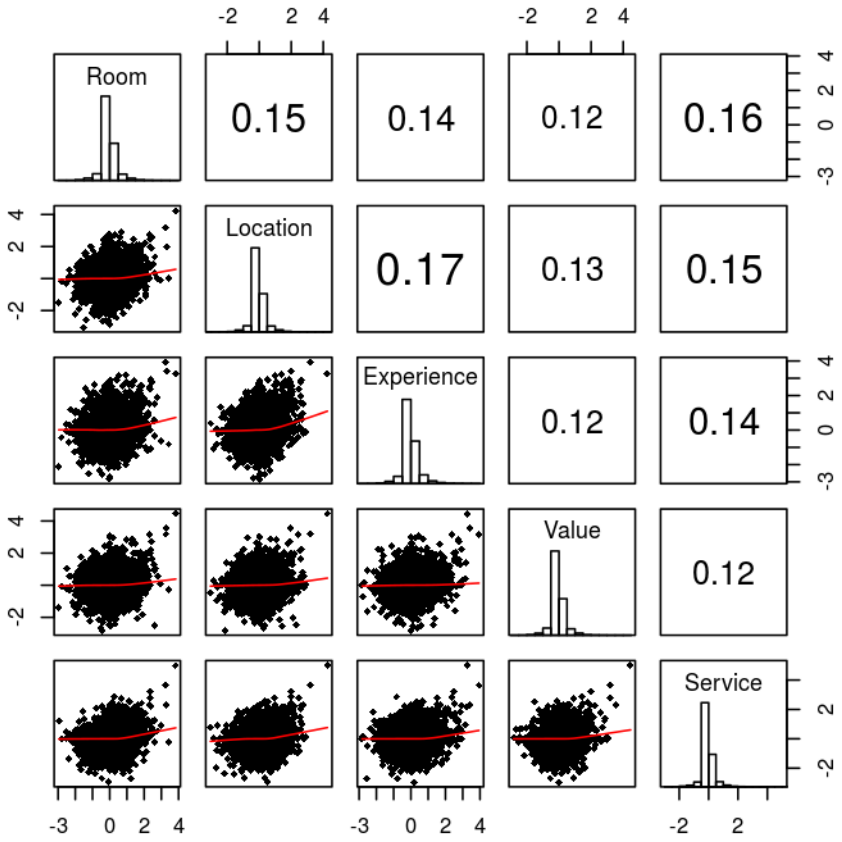

Figure 6: Correlation between $\widehat{\boldsymbol{\beta}}$ values.

The estimated mean vector $\widehat{\boldsymbol{\mu}}$ is reported in Table 6.

Room	Service	Experience	Location	Value
-0.216	0.246	0.126	0.021	-0.178

Table 6: Estimated $\widehat{\boldsymbol{\mu}}$.

We can notice that the estimated $\widehat{\boldsymbol{\mu}}$ values for the aspects *Service* and *Experience* have a higher mean while *Room* and *Value* have a lower one.

The $\widehat{\boldsymbol{\Sigma}}$ covariance matrix values are reported in Table 7. The estimated $\widehat{\boldsymbol{\Sigma}}$ have higher variability values for *Service*, *Experience* and *Location* while they have lower values for *Room* and *Value*.

	Room	Service	Experience	Location	Value
1	5.18	-3.00	-0.47	-1.31	-0.41
2	-3.00	23.85	-10.23	-8.29	-2.33
3	-0.47	-10.23	17.49	-5.47	-1.31
4	-1.31	-8.29	-5.47	16.01	-0.95
5	-0.41	-2.33	-1.31	-0.95	5.00

Table 7: Estimated $\widehat{\boldsymbol{\Sigma}}$.

5.3 Document-level parameters

The next step is to summarize the estimates of the document-level parameters, namely the aspect ratings $\widehat{S}$ and the aspect weights $\widehat{\boldsymbol{\alpha}}$. The empirical distribution of the latter values are displayed in Figure 7. We notice that the medians of each aspect weight are roughly aligned around the 0.1 value, with some notable differences. There are several outlying values covering the entire range from 0.5 to 1. Notice that the estimated $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ reported in Tables 6 and 7 refer to the $\boldsymbol{\eta}$ transformation of the aspect weight (see equation 10), yet higher values of the mean and covariance of $\boldsymbol{\eta}_d$ are reflected in the corresponding component of $\boldsymbol{\alpha}_d$: this is apparent by comparing the figure with the two aforementioned tables. Indeed, the mean values are higher for the *Service* and the *Experience* aspects while they are lower for *Room* and *Value*. Likewise, the variability is higher in *Service* and *Experience* and less in *Room* and *Value*.

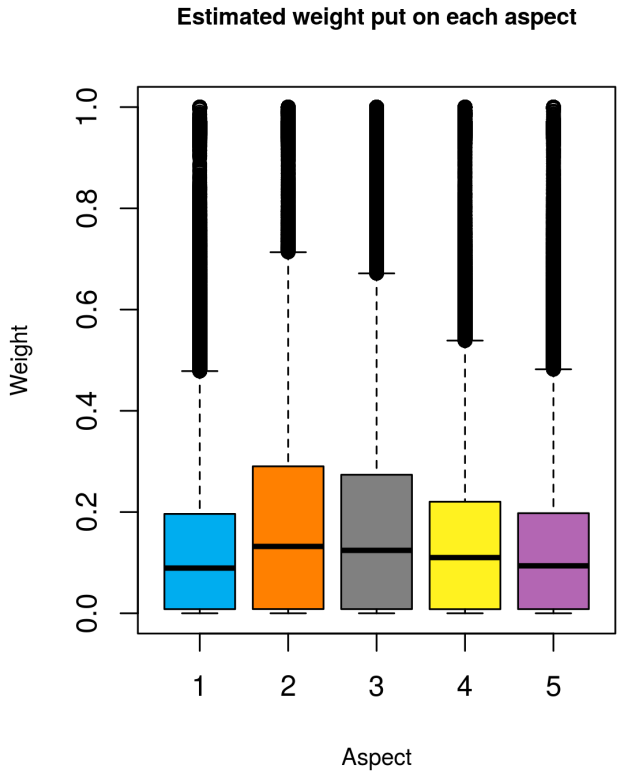

Figure 7: Empirical distribution of estimated aspect weights.

The estimated aspect ratings $\widehat{\mathbf{s}}$ are displayed in Figure 8. The medians for the *Room* and *Value* aspects are close to the value 1, whereas the medians for the *Service* and *Experience* aspects are close to 4 and the median of the *Location* aspect is near the value 3. This directly mirrors a correspondig ranking in the aspect rating given by reviewers. It would be possible to transform the estimated aspect ratings by mapping them in the same range of values adopted by users for the overall review, and this could give some more interpretable results. This point will be considered in the third article of this thesis.

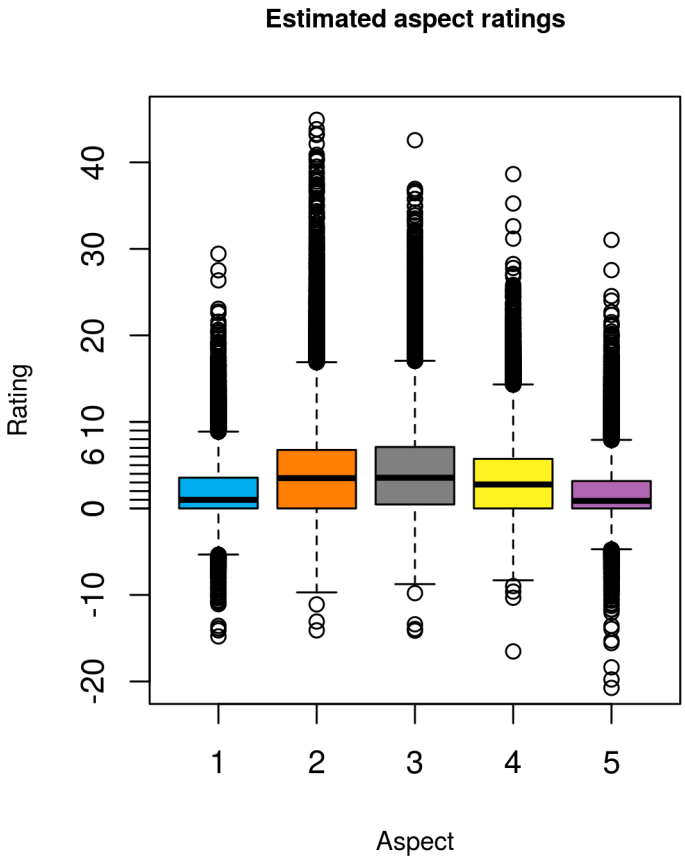

Figure 8: Empirical distribution of estimated aspect ratings.

5.4 Example of algorithm application

We have applied both steps of the algorithm and picked up one of the reviews. After the first step, the topics are identified, each sentence is associated to one topic and the result is displayed underneath. The color code for aspects is the same adopted previously.

"For the price we paid using Priceline, the location of the hotel was definitely worth it. Though the rooms were quite small compared to other Marriotts, we were within walking distance to Chinatown and close to a T Station that went to the Museum of Fine Arts and Prudential Tower. The staff members were pleasant enough. There was free cucumber and orange water available in the lobby throughout the day. In the morning, hot coffee and water was provided. One major con: the walls are very thin. Throughout the night and morning, I could hear people in the hallway, neighbors, and beeping cars outside even though I was on the 6th floor."

After the application of the latent regression part, we obtain the estimates for the aspect rating and the aspect weights. By comparing them to the identified topics, we notice there is a close correspondence between the rates and weight assigned by the

algorithm and the text. Also the overall rating given by the reviewer, 3, is similar to the fitted one, 3.13.

	Room	Service	Experience	Location	Value	Overall Rating
Aspect Rating	1	3	0	4	4	Given = 3
Aspect Weight	21.82%	10.73%	12.14%	16.18%	39.12%	Fitted = 3.13

Table 8: LRR application, example for a review.

In Figure 9 we plot the fitted values $\widehat{\mathbf{r}}$ versus the observed ones, by doing separate boxplots of all the estimated ratings corresponding to a given value of the overall rating. We notice how the median of the fitted data closely corresponds to the observed rating, and indeed the correlation coefficient between the two variables is equal to 0.88.

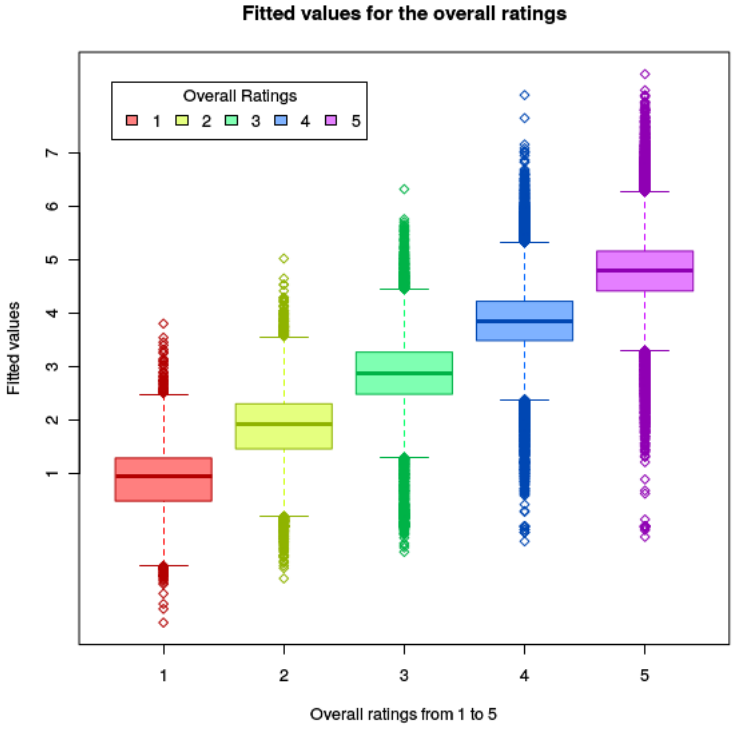

Figure 9: Estimated vs observed overall rating. The color codes used in this graph are employed to underline the difference between the ratings and have no connection to the colors used in the previous images.

6 Conclusion

The application of our model to different text review contexts should be tested with further data in order to be able to draw more reliable conclusions. To this end, in the next article of this thesis we apply the algorithms to a benchmark dataset containing the overall ratings as well as the individual ratings. The analysis of the aspect ratings can be

the harbinger of a multitude of application tasks such as entity ranking based on aspect ratings, analysis of reviewers rating behavior in a timespan or review summarization.

Appendix: Computational details

Here we focus in particular on the LRR step of the proposed method, which has required an ad-hoc implementation based on the `R` software. In particular, for the estimation of document-specific parameters given by equation (16), the `alabama` (Varadhan, 2015) package has been employed for constrained optimization, and the `TMB` package (Kristensen et al., 2016) for speeding up the evaluation of the objective function using `C++`. The optimization has been made fast by deploying suitable parallel processing using the `parallel` package, which is part of the standard `R` distribution. For what concerns the corpus-level parameters, instead, the key task has been the usage of suitable sparse matrices, due to the nature of the data analyzed. Two different sparse matrix implementations have been used in different parts of the code, provided by the `Matrix` (Bates and Maechler, 2017) and `SparseM` (Koenker and Ng, 2017) packages respectively. Finally, ridge regression has been applied by using the powerful `LiblineaR` package (Helleputte, 2017), which is the `R` port of the `LIBLINEAR C/C++` library for machine learning.

References

Aitchison, J. (1986). *The Statistical Analysis of Compositional Data*. Chapman and Hall London.

Balikas, G., Amini, M.-R., and Clausel, M. (2016). On a topic model for sentences. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 921–924. ACM.

Bates, D. and Maechler, M. (2017). *Matrix: Sparse and Dense Matrix Classes and Methods*. R package version 1.2-10.

Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120. ACM.

Blei, D. M. and Lafferty, J. D. (2009). Topic models. *Text Mining: Classification, Clustering, and Applications*, 10(71):34.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Büschken, J. and Allenby, G. M. (2016). Sentence-based text analysis for customer reviews. *Marketing Science*, 35(6):953–975.

Chen, R.-C., Swanson, R., and Gordon, A. S. (2016). An adaptation of topic modeling to sentences. *CoRR*, abs/1607.05818.

Cui, H., Mittal, V., and Datar, M. (2006). Comparative experiments on sentiment classification for online product reviews. In *AAAI*, volume 6, pages 1265–1270.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web*, pages 519–528. ACM.

Devitt, A. and Ahmad, K. (2007). Sentiment polarity identification in financial news: A cohesion-based approach. In *ACL*, volume 7, pages 1–8.

Efron, B. and Hastie, T. (2016). *Computer Age Statistical Inference*, volume 5. Cambridge University Press.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The Elements of Statistical Learning*, volume 1. Springer Series in Statistics New York.

Gelman, A., Bois, F., and Jiang, J. (1996). Physiological pharmacokinetic analysis using population modeling and informative prior distributions. *Journal of the American Statistical Association*, 91(436):1400–1412.

Goldberg, A. B. and Zhu, X. (2006). Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pages 45–52. Association for Computational Linguistics.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

Griffiths, T. L., Steyvers, M., Blei, D. M., and Tenenbaum, J. B. (2005). Integrating topics and syntax. In *Advances in Neural Information Processing Systems*, pages 537–544.

Helleputte, T. (2017). *LiblineaR: Linear Predictive Models Based on the LIBLINEAR C/C++ Library*. R package version 2.10-8.

Hornik, K. and Grün, B. (2011). topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, 40(13):1–30.

Jagarlamudi, J., Daumé III, H., and Udupa, R. (2012). Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213. Association for Computational Linguistics.

Jiang, J. (2007). *Linear and Generalized Linear Mixed Models and their Applications*. Springer Series in Statistics New York.

Koenker, R. and Ng, P. (2017). *SparseM: Sparse Linear Algebra*. R package version 1.77.

Kristensen, K., Nielsen, A., Berg, C. W., Skaug, H., and Bell, B. M. (2016). TMB: Automatic differentiation and Laplace approximation. *Journal of Statistical Software*, 70(5):1–21.

Lafferty, J. D. and Blei, D. M. (2006). Correlated topic models. In *Advances in Neural Information Processing Systems*, volume 18, pages 147–154.

Lee, Y. and Nelder, J. A. (1996). Hierarchical generalized linear models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 619–678.

Lee, Y. and Nelder, J. A. (2009). Likelihood inference for models with unobservables: another view. *Statistical Science*, pages 255–269.

Lee, Y., Nelder, J. A., and Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. CRC Press Boca Raton.

Lu, Y., Zhai, C., and Sundaresan, N. (2009). Rated aspect summarization of short comments. In *Proceedings of the 18th International Conference on World Wide Web*, pages 131–140. ACM.

MacKay, D. J. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

McAuley, J. and Leskovec, J. (2013). Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM.

Mcauliffe, J. D. and Blei, D. M. (2008). Supervised topic models. In *Advances in Neural Information Processing Systems*, pages 121–128.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models, Second Edition*. Chapman and Hall/CRC Boca Raton. Chapman & Hall.

Melián-González, S., Bulchand-Gidumal, J., and González López-Valcárcel, B. (2013). Online customer reviews of hotels: As participation increases, better evaluation is obtained. *Cornell Hospitality Quarterly*, 54(3):274–283.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization, Second Edition*. Springer-Verlag New York.

Ormerod, J. T. and Wand, M. P. (2010). Explaining variational approximations. *The American Statistician*, 64(2):140–153.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124. Association for Computational Linguistics.

Qiu, Z., Wu, B., Wang, B., Shi, C., and Yu, L. (2014). Collapsed Gibbs sampling for latent Dirichlet allocation on spark. In *Proceedings of the 3rd International*

Conference on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, volume 36, pages 17–28. JMLR. org.

Ramesh, A., Goldwasser, D., Huang, B., Daume, H., and Getoor, L. (2014). Understanding mooc discussion forums using seeded lda. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 28–33.

Rao, C. R. (1973). *Linear Statistical Inference and its Applications*, volume 2. Wiley New York.

Robinson, G. K. (1991). That BLUP is a good thing: the estimation of random effects. *Statistical Science*, 6:15–32.

Smyth, G. K. (1996). Partitioned algorithms for maximum likelihood and other non-linear estimation. *Statistics and Computing*, 6(3):201–216.

Snyder, B. and Barzilay, R. (2007). Multiple aspect ranking using the Good Grief algorithm. In *HLT-NAACL*, pages 300–307.

Srivastava, M. S. and von Rosen, D. (2002). Regression models with unknown singular covariance matrix. *Linear Algebra and its Applications*, 354(1-3):255–273.

Titov, I. and McDonald, R. (2008). Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web*, pages 111–120. ACM.

Tutz, G. (2011). *Regression for Categorical Data*, volume 34. Cambridge University Press.

Varadhan, R. (2015). *alabama: Constrained Nonlinear Optimization*. R package version 2015.3-1.

Vásquez, C. (2011). Complaints online: The case of TripAdvisor. *Journal of Pragmatics*, 43(6):1707–1717.

Wang, C., Blei, D., and Heckerman, D. (2012). Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298*.

Wang, H., Lu, Y., and Zhai, C. (2010). Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 783–792.

Wang, H., Lu, Y., and Zhai, C. (2011). Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 618–626.

Zhai, C. and Massung, S. (2016). *Text Data Management and Analysis: a Practical Introduction to Information Retrieval and Text Mining*. Morgan & Claypool New York.

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., and Li, X. (2011). Comparing Twitter and traditional media using topic models. In *Proceedings of the 33rd European Conference on Information Retrieval*, pages 338–349. Springer-Verlag Berlin.

Evaluation and Practical Application of Latent Aspect Rating Analysis

Abstract

In this article we focus on a comparative application of the algorithm developed in "Latent Aspect Rating Analysis: a Model-Based Approach", the second article of this thesis. To demonstrate the effectiveness of our method, we make a comparison using a dataset containing benchmark data. The comparison is done by applying the two steps of the aforementioned algorithm. Some practical applications to marketing are illustrated.

Key words: Latent rating analysis, LDA, Marketing, Opinion analysis, Time series.

1 Introduction

When faced to making a decision such as which product to buy or which service to use, we often rely on opinions of persons we know and whom we trust. Nowadays, these opinions are widespread on the Internet under the form of reviews generated by unknown users, whom we trust if we encounter the same opinion shared by many reviewers. Reading and using such a large amount of data is a difficult task and it is even harder to interpret. This type of User-Generated Content represents subjective opinions and it has a basic structure: who writes the opinion (the user), what it is about (a product or a service), and the opinion content. From the opinion content we mine the needed information. Opinion mining is important for three major reasons and it is useful for products or service providers and for customers alike. From the supplier's point of view, one reason is that knowing customer preferences we can optimize a product or intervene with targeted advertising. The second reason, useful for business intelligence, is that by aggregating reviews from many users we can assess a more general opinion. In this manner manufacturers can know where their products have advantages or disadvantages over the competitors. The third reason, and this is the customer's point of view, is to aid optimize decisions, such as choosing a product to buy or a service to use. Reading many reviews is usually difficult and confusing so it

is necessary to automatize processes that will allow us to extract useful information. Marketing people have limited time available and want to make decisions based on summarized data that give insightful information.

By means of the application of the algorithm presented in the "Latent Aspect Rating Analysis: a Model-Based Approach" article of this thesis, we propose a method to extract meaningful information from unstructured data in order to suit the aforementioned needs. In this article we try to assess the goodness of our algorithm by comparing the results obtained through two similar methodologies. Initially we have downloaded the `Java` code used by Wang, Lu, and Zhai (2010) for their algorithm. After running it, we realized that the code made available does not provide document-level parameters but just aggregate estimates which are difficult if not impossible to interpret. Thus the direct comparison with their method resulted unfeasible, but we use their dataset as benchmark data.

Wang et al. (2010) obtain their results applying a two-step algorithm at the review level. In the first step they segment the reviews into aspects using a set of seed words and a suitable segmentation algorithm. In the second step they apply the method of Latent Rating Regression (LRR) (Wang et al., 2010; Wang, Lu, & Zhai, 2011) to estimate the rating values for the aspects, the aspect weights and the overall rating. Here we employ a two-step algorithm as well. In particular, at the first step we employ Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003) to obtain the aspects, and at the second step we use our enhanced version of LRR. Details of the latter method are presented in the second article of this thesis.

The first step, in our case, is dedicated to the topics segmentation and this can be developed in four different alternative approaches, as follows.

1. LDA applied to entire reviews;
2. LDA applied to entire reviews influenced by a set of seed words;
3. LDA applied to reviews divided into sentences;
4. LDA applied to reviews divided into sentences influenced by a set of seed words.

The seed words employed in 2. and 4. were the same used by Wang et al. (2010) in their article. It will be shown that seeded version are more suitable for applying LRR afterwards. From

now on we adopt the term `base` to indicate the method that uses LDA applied to entire reviews at the first step, and `sentence` the method that uses LDA applied to review corresponding to sentences at the first step, both followed by LRR.

The structure of this article is as follows. In the second section we investigate the data gathering procedure and we specify the data manipulation in order to obtain structured workable data sets. In the third section, we underline the limitations of the benchmark data. In the fourth section we apply the first step of our algorithm based on LDA searching for seven aspects and we focus on the outcomes of the four different approaches. In the fifth section of the paper we apply the second part of the algorithm, the latent rating regression, to the previously modeled data and compare the results of `base` and `sentence` with the existing benchmark data. In the sixth section we provide some time series plots to the estimated data and compare them with the benchmark data. Finally, a discussion on the best method is assessed by computing RMSE of prediction for each of the hotels of the benchmark data. The last section is dedicated to the conclusions.

2 Data gathering and manipulation

The considered dataset was obtained by the authors of the LARA algorithm (Wang et al., 2010) between 01/01/2002 and 12/01/2009. It consists of various reviews of hotels which were collected from the travel-related website `TripAdvisor` and can be found at the following `url`: `http://sifaka.cs.uiuc.edu/~wang296/Data/LARA/TripAdvisor/Review_Texts.zip`. Due to the longitudinal data gathering, the information is slightly different from one time interval to another. For example, in some cases the price metadata is missing, other reviews have the reviewer's location metadata missing, and so on. The format of the files found inside the `Review_Texts` data archive is `dat`. The total number of files corresponding to the number of hotels considered is 1 759 and the total number of reviews is 132 257. In each file there is a three row information header consisting of the hotel's overall rating, the average price and an URL. The URL sequence contains the name of the hotel and the city location. Then follows the main information we need to process for our analysis which is the overall rating of the review and the review itself. Further metadata contribute essential information. The metadata consists of: `Author`, `Date`, `Number of Reader`, `Number of Helpful Judgement`, `Rooms`, `Location`, `Cleanliness`, `Check in / Front Desk`, `Service and Business Service`. The last seven metadata are the aspects we

are interested in. The value range for the aspects is between 1 and 5. A value of -1 indicates a missing value. In the previously specified period, `TripAdvisor` divided the aspects differently than nowadays. The subdivision was a result of the company need to investigate the *Business Service* aspect. Smartphones and laptop computers were not common and were not very fast in the years 2002-2009, thus a dedicated service for the business people had to be offered by hotels. We will use the described benchmark data to compare it with the aspects and the ratings we have detected and computed during the application of our algorithm. For processing reasons we have to render the data uniform. The aim is to create a table-like structure that would be easy to use. As an example of a composition of a `dat` file, see an excerpt in Appendix A. We created a `bash` ("Free Software Foundation. Bash (4.3.48(1) [Unix shell program].", 2017) script in the Linux operating system in order to parse all the files and remove the unnecessary or incomplete data. In order to accomplish this, we follow the following steps.

- Remove all the information before the first *<Author>* tag occurrence;
- Remove all the uninformative lines such those commencing by *<img*, *<No. Reader>* and *<No. Helpful>* because they contain non-usable information for our purpose;
- Substitute the blank spaces in *<Check in / front desk>* and *<Business service>* with an underscore symbol because different operating systems interpret blank space in a different manner;
- Add a tab space as a delimiter after the *<Author>*, *<Content>*, *<Date>*, *<Overall>*, *<Value>*, *<Room>*, *<Location>*, *<Cleanliness>*, *<Check_in/front_desk>*, *<Service>* and *<Business_service>* tags in order to set the `dat` file compatible with the `csv` format;
- We notice that every review is accompanied by exactly 11 repeated fields. We split thus each `dat` file into multiple `csv` files, each of them containing only one set of fields;
- We perform a transposition of the file content in order to have the names of the fields on top and the data underneath with the aid of the command line `csvtool` software;
- We change encoding into ISO-8859-1 in order to capture the West European characters.

After performing the above operations, we have reduced the usable data to 1 118 hotels for 123 042 reviews.

3 Limitations of the benchmark data

One important limitation for the benchmark dataset is that it contains a high percentage of non-available (NA) data. This missing data are not homogeneous but they are distributed with different proportions over all the aspects. The percentage of the NA data is reported in Table 1. In the dataset, there are 28 108 reviews that contain NA values in all the aspects, and 50 841 reviews contain from 1 to 6 aspects with missing data. The remaining 44 093 reviews contain complete data.

	Value	Room	Location	Cleanliness	Check_in / front_desk	Service	Business service
NA	23.67%	22.72%	41.74%	22.72%	41.69%	24.23%	61.37%

Table 1: Percentages of NA data for each aspect.

Another limitation of the benchmark dataset is the common pattern behavior when users assign aspect ratings. If they are keen on an aspect, they tend to inflate and assign the same value to all the other aspects. This happens regardless of if there was a discrepancy between the sentiment words used to describe an aspect and the assigned value. In marketing literature this rating bias is a known behavioral phenomenon, and it is called the halo effect (e.g. Beckwith, Kassarjian, & Lehmann, 1978). Having the same value for all the variables may produce strong correlation between the aspect ratings, resulting in information redundancy. In order to illustrate this fact we apply a Principal Component Analysis (PCA) procedure to the dataset formed by all the aspect reviews for the benchmark data, obtaining one important dimension only; see Figure 1.

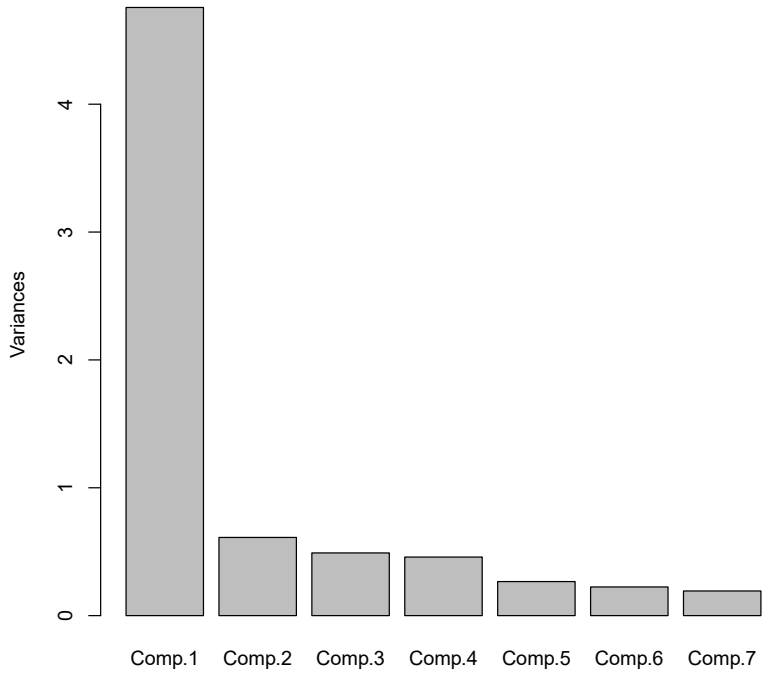

Figure 1: Principal components analysis applied to benchmark data.

The last limitation in using this benchmark dataset is that seven aspects are not appropriate for this kind of analysis. In fact, `TripAdvisor` has reduced and changed in the past years the number of aspects for some touristic structures to six (*Value*, *Rooms*, *Location*, *Cleanliness*, *Service*, *Sleep Quality*) suggesting that it may be difficult to completely identify seven aspects. Due to the previous considerations, the dataset is not ideal, making the benchmarking challenging.

4 LDA application

In this section, we present the preparation of the datasets that can be compared with the benchmark data. After the files preparation, we import the data in the `R` (R Core Team, 2017) environment. We create then an alternative version of the same dataset, having thus two datasets available. One is the original dataset that contains the full reviews, and another one contains the reviews subdivided into sentences obtained with the `StanfordCoreNLP` (Hornik, 2017) package.

Then, using a procedure similar to the one described in the first article of this thesis, we pre-process the data in order to obtain a clean corpus for each dataset. We apply thus the first step of our algorithm, LDA, to the first dataset processing the whole reviews. The parameters were estimated and inferred with the aid of LDA employing the Gibbs sampling technique by means of the `lda` function of the `topicmodels` package. In particular, throughout this article, for the LDA application, in the `control` argument of the lda function, the slots `burnin` and `iter` (for the `method = "Gibbs"`) were set to 3000. The former represents the number of discarded initial Gibbs iterations and the latter represents the number of subsequent draws. In order to match the number of topics of the benchmark data, we fix the number of latent aspects to seven. In Table 2 we report the 20 most frequent words of the discovered topics.

	Topic_1	Topic_2	Topic_3	Topic_4	Topic_5	Topic_6	Topic_7
1	beach	just	staff	stay	room	room	locat
2	resort	get	great	room	bed	day	great
3	food	like	stay	dollar	bathroom	arriv	walk
4	pool	will	love	nice	floor	one	stay
5	restaur	realli	friend	view	night	ask	good
6	drink	can	servic	night	small	call	clean
7	bar	time	wonder	park	door	servic	breakfast
8	day	want	excel	price	shower	book	street
9	water	thing	best	area	two	told	citi
10	good	place	recommend	place	one	frontdesk	help
11	buffet	peopl	help	rate	coffe	hour	comfort
12	vacat	one	enjoy	pool	larg	check	minut
13	also	back	return	restaur	size	even	staff
14	time	review	beauti	free	nice	anoth	shop
15	beauti	good	time	servic	lobbi	guest	block
16	peopl	say	well	also	breakfast	time	recommend
17	alway	got	new	great	use	manag	valu
18	tip	much	experi	old	tvset	wait	room
19	get	nice	perfect	good	also	problem	friend
20	kid	littl	fantast	day	nois	said	restaur

Table 2: 20 most frequent terms resulted after the LDA application to entire reviews for 7 topics.

After the LDA application, the topics result unclear. Unlike in the previous example from the "Latent Aspect Rating Analysis: a Model-Based Approach" where we used 5 topics which were identified with ease, here we can empirically barely label three of them. Topic_7 could be labeled as *Location*, Topic_5 as *Room* and Topic_4 as *Value*. The other topics are harder to identify due to the mixture of the words that do not seem pertinent to a pre-defined aspect label. For this reason we discard the results obtained with the previous method from further processing and attempt a different approach.

Wang et al. (2010) applied the segmentation step in order to discover aspects employing a set of predefined seed words (see the previous article). We use a different approach to identify the aspects with the aid of the seeded words added to LDA (Jadeja & Pandya, 2014; Jagarlamudi, Daumé III, & Udupa, 2012). Based on the aforementioned literature, the application of the latter algorithm can give better results than plain LDA (Blei et al., 2003). We have to supply the number of topics and a set of significant words for each topic. In this case we chose, for each aspect, the same four leading words chosen by Wang et al. (2010):

- *Value*: value, price, quality, worth;
- *Room*: room, suite, view, bed;
- *Location*: location, traffic, minute, restaurant;
- *Cleanliness*: clean, dirty, maintain, smell;
- *Check_ in/front_ desk*: stuff, check, help, reservation;
- *Service*: service, food, breakfast, buffet;
- *Business service*: business, center, computer, internet.

We have applied again, to the first dataset, LDA with the Gibbs sampling method and seeded words. We can notice in Table 3 that the twenty most frequent words defining each discovered aspect are semantically coherent. The topics that are best characterized by the grouped words are *Room*, *Location* and *Check_ in/front_ desk*. The others seem to present mixed elements that can belong to any other aspect. We keep this dataset for further processing with the second step of our algorithm.

	Value	Room	Location	Cleanliness	Check_in/ front_desk	Service	Business service
1	staff	room	good	just	room	beach	stay
2	great	bed	locat	get	day	resort	great
3	stay	floor	breakfast	like	one	food	nice
4	love	view	great	will	arriv	pool	dollar
5	servic	bathroom	walk	realli	ask	day	locat
6	friend	night	stay	can	call	drink	room
7	wonder	small	staff	want	servic	bar	park
8	best	one	restaur	time	check	restaur	night
9	enjoy	door	clean	thing	hour	water	street
10	beauti	stay	minut	one	book	buffet	walk
11	time	shower	room	place	told	good	place
12	return	two	help	peopl	even	also	price
13	excel	suit	excel	back	frontdesk	vacat	free
14	experi	look	recommend	review	staff	dollar	car
15	recommend	nois	friend	much	reserv	alway	area
16	will	nice	also	got	anoth	kid	busi
17	year	tvset	well	say	guest	peopl	san
18	trip	size	citi	clean	time	time	comfort
19	well	area	book	good	manag	get	block
20	perfect	larg	shop	think	problem	tip	clean

Table 3: 20 most frequent terms resulted after the LDA application to entire reviews influenced by a set of seed words for 7 topics.

We employed then the second dataset formed by the reviews subdivided into sentences, and we applied the same LDA model as in the previous cases but constraining the sentences of each review to hold only one topic. This was accomplished by keeping the α parameter of the symmetric Dirichlet distribution at a very low value (0.001). The twenty most frequent words obtained for each topic after the application of the procedure are reported in Table 4.

	Value	Room	Location	Cleanliness	Check_in/ front_desk	Service	Business service
1	stay	room	walk	room	room	food	beach
2	resort	bed	locat	view	staff	restaur	pool
3	great	clean	shop	stay	get	breakfast	day
4	time	bathroom	minut	great	help	good	get
5	just	shower	great	night	day	buffet	resort
6	review	nice	restaur	floor	time	drink	bar
7	will	water	street	dollar	one	bar	one
8	night	one	beach	locat	dollar	great	water
9	place	comfort	get	nice	servic	servic	night
10	back	small	just	good	arriv	eat	great
11	return	day	area	servic	friend	one	nice
12	trip	well	away	price	will	get	time
13	one	door	right	book	call	also	peopl
14	year	floor	close	clean	ask	like	area
15	vacat	like	block	staff	frontdesk	day	beauti
16	good	towel	can	one	peopl	dinner	like
17	punta	tvset	also	pool	even	room	can
18	travel	larg	place	suit	make	night	chair
19	cana	size	good	ocean	told	even	just
20	week	also	take	get	hour	ate	also

Table 4: 20 most frequent terms resulted after the LDA application to reviews divided into sentences for 7 topics.

Topics *Room*, *Location* and *Service* seem to be composed by coherent terms. The other topics seem to have elements that can be ascribable to *Value*, *Cleanliness* and *Check in / front desk* although they contain some words that are not coherent with the assigned label. The *Business service* label was assigned to the last topic because it was the last one available, but it is composed of words unrelated to this aspect. Due to these uncertainties we discard these results obtained by the application of the LDA model to the reviews subdivided into sentences.

We apply LDA again, as in the previous case, but, in addition, we use the seed words set we

have already described. The results, apparently, seem to have the best representation so far. We report the twenty most frequent words in Table 5.

	Value	Room	Location	Cleanliness	Check_in/ front_desk	Service	Business service
1	stay	room	locat	room	staff	food	dollar
2	great	bed	walk	beach	room	breakfast	internet
3	resort	view	restaur	pool	help	buffet	day
4	time	suit	minut	clean	reserv	restaur	busi
5	price	clean	great	day	check	servic	night
6	night	nice	traffic	water	friend	good	get
7	just	floor	beach	smell	get	drink	center
8	review	bathroom	shop	get	servic	bar	comput
9	will	comfort	street	dirti	time	great	one
10	place	small	area	one	one	room	resort
11	valu	great	just	maintain	arriv	one	also
12	good	stay	right	time	day	eat	beach
13	worth	one	away	towel	frontdesk	get	time
14	return	size	close	nice	call	day	can
15	back	larg	block	like	peopl	also	room
16	qualiti	good	get	night	ask	like	will
17	one	well	good	chair	make	dinner	take
18	trip	pool	place	just	will	night	free
19	year	area	also	even	even	even	show
20	vacat	two	can	area	stay	coffe	park

Table 5: 20 most frequent terms resulted after the LDA application to reviews divided into sentences influenced by a set of seed words for 7 topics.

The last result was kept for further processing. We selected thus the whole-review and the sentence-based models obtained through the application of LDA with the seed words. The comparison of the models with the benchmark data is detailed in the next section.

5 LRR application

The results obtained after the LDA application are saved in a matrix form. On these matrices we apply our Latent Rating Regression algorithm and we obtain the estimated aspect ratings. For the computational details and the structure of the matrices see the previous article in this thesis. We test first the relationship between the variables representing the aspects of the `base` dataset and those of the benchmark data. In particular, we are interested in the polarity of the relationship and its strength. We apply a statistical correlation procedure between the `base` (estimated) data and the (observed) benchmark data on each considered aspect. The correlation was applied to the hotels aggregated. There are thus 1 118 points to be displayed and we use scatterplots in order to accomplish this. In order to distinguish the hotels which have the three highest number of reviews, we identified them with a different color. All names have been changed to protect the privacy of the legal owners. With cyan we encoded the "Noctilucent" hotel, with magenta we encoded the "Cirrus" hotel and with yellow we encoded the "Cumulus" hotel. The first hotel counts 2 308 reviews, the second 1 558 and the third 1 421. All the other hotels are represented by black circles. The results are displayed in Figure 2.

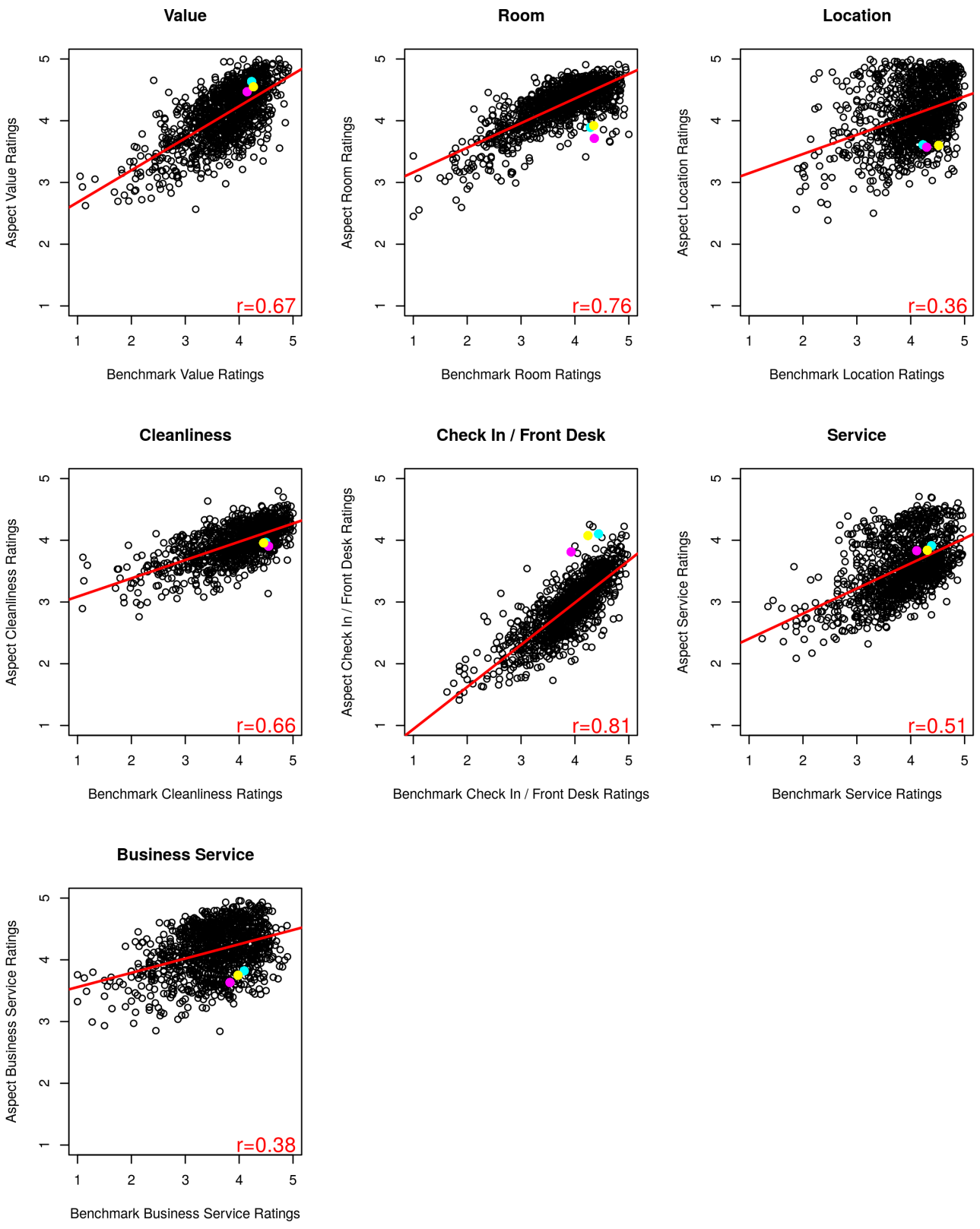

Figure 2: Estimated `base` ratings versus benchmark ratings for all the aspects.

The correlation coefficient r, displayed on each aspect graphic, indicates how the variables are linearly related. The slope is positive so this indicates the variables are positively related. Some of the aspects as *Check In / Front Desk* or *Room* seem to be strongly correlated while *Location* and *Business Service* have a moderate degree of correlation, but the results are generally satisfactory.

We apply the same criterion to the `sentence` data and we obtain positive correlations also in this case, with somewhat stronger relationships between the aspect variables and the benchmark data in all the aspects except for *Location*, see Figure 3.

To assess which of the two algorithms perform better, we report in Table 6 the correlations for each aspect and each dataset. The figure underlined represents the higher one. By looking at the table we can conclude that globally, the algorithm applied to `sentence` yields higher correlation coefficients than the algorithm applied to `base`.

	Value	Room	Location	Cleanliness	Check_in / front_desk	Service	Business service
`base`	0.67	0.76	0.36	0.66	<u>0.81</u>	0.51	0.38
`sentence`	<u>0.73</u>	<u>0.78</u>	<u>0.42</u>	<u>0.71</u>	0.76	<u>0.81</u>	<u>0.66</u>

Table 6: Correlation values for `base` and `sentence` datasets.

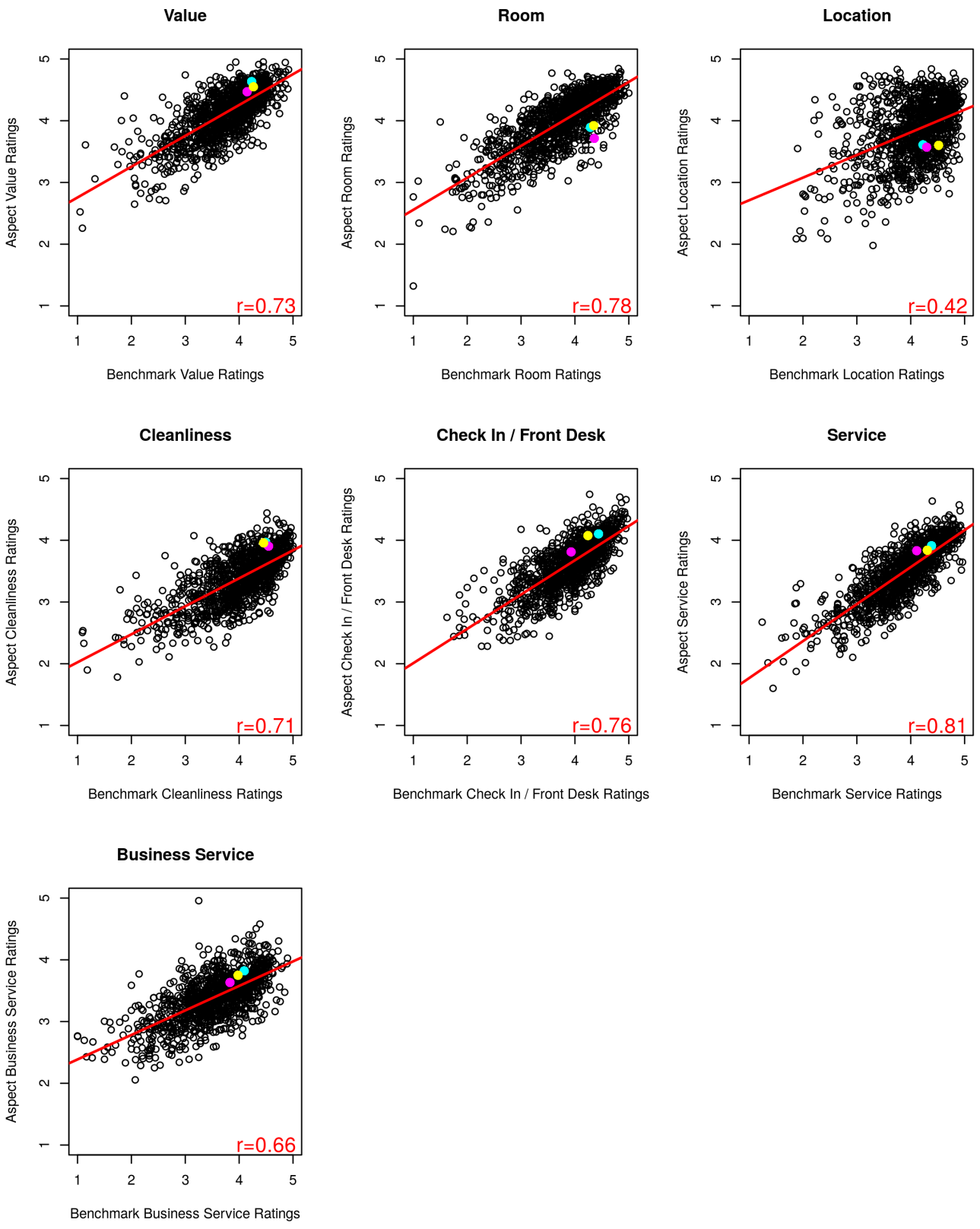

Figure 3: Estimated `sentence` ratings versus benchmark ratings for all the aspects.

6 Time series application

The salient latent aspects which are derived in a data-driven way by analyzing texts of online service reviews can be employed to supply further information over a period of time. A practical application of the algorithm can be useful in marketing research, especially for decision making. The modification in time of the ratings of the aspects for the same hotel can give an insight on the goodness of the policies applied at management level. This information can also be used in order to maximize the quality of the services offered. We apply the time series for the *Overall* ratings as well as for all the other aspects in order to compare the estimated ratings obtained with the two approaches, `base` and `sentence` to the benchmark data. We chose the three most reviewed hotels from the benchmark dataset. In all the following pictures regarding the aspects, the benchmark dataset is encoded by a black color, the estimated `base` review level is encoded by red and the `sentence` review level is encoded by green. The blue line represents the monthly average of daily number of reviews.

6.1 Overall ratings

After the application of the time series to the *Overall* ratings, for the "Noctilucent" hotel, we notice that both `base` and `sentence` estimated ratings follow roughly the benchmark data, in some portions overlapping it. This happens even in the initial part of the series when the reviews available are few. The `sentence` estimated data seem to be closer to the benchmark data. (See Figure 4).

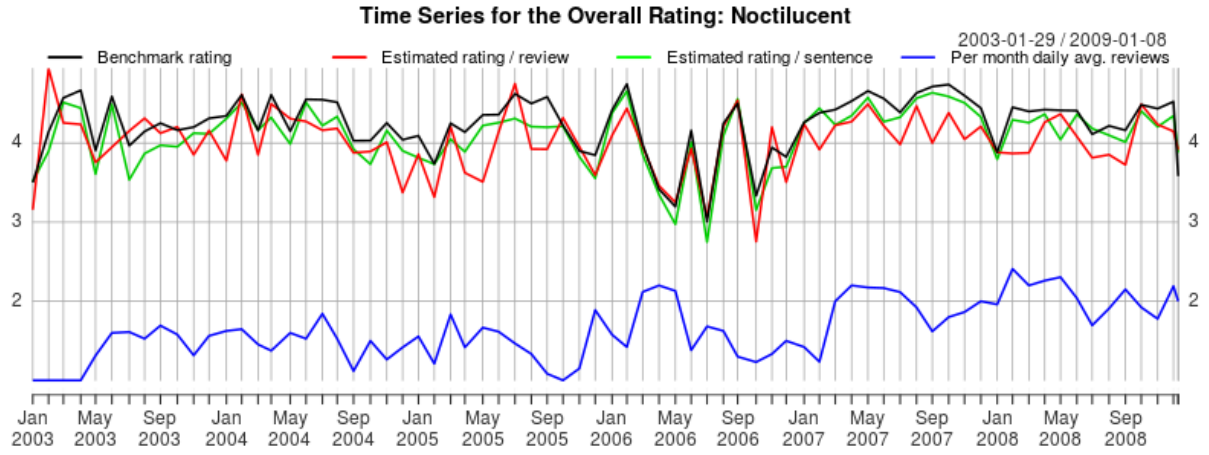

Figure 4: Time series for the *Overall* rating of the Noctilucent hotel.

In the case of the *Overall* ratings, for the "Cirrus" hotel, the trend is similar to the one observed in the previous case. (See Figure 5).

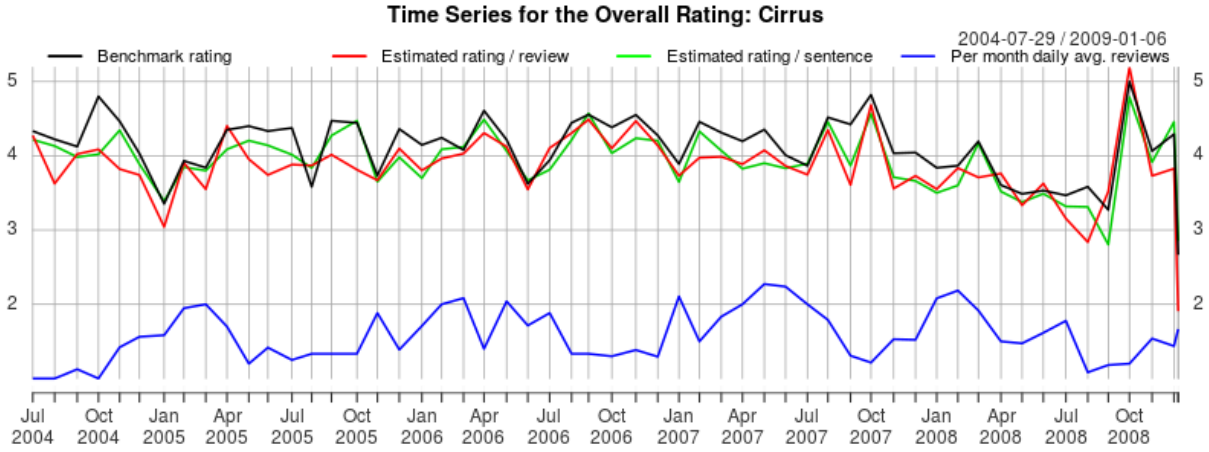

Figure 5: Time series for the *Overall* rating of the Cirrus hotel.

In the case of the *Overall* ratings, for the "Cumulus" hotel, the trend changes. The ratings seem to be underestimated for both `base` and `sentence`, no matter if the number of reviews increases. (See Figure 6.)

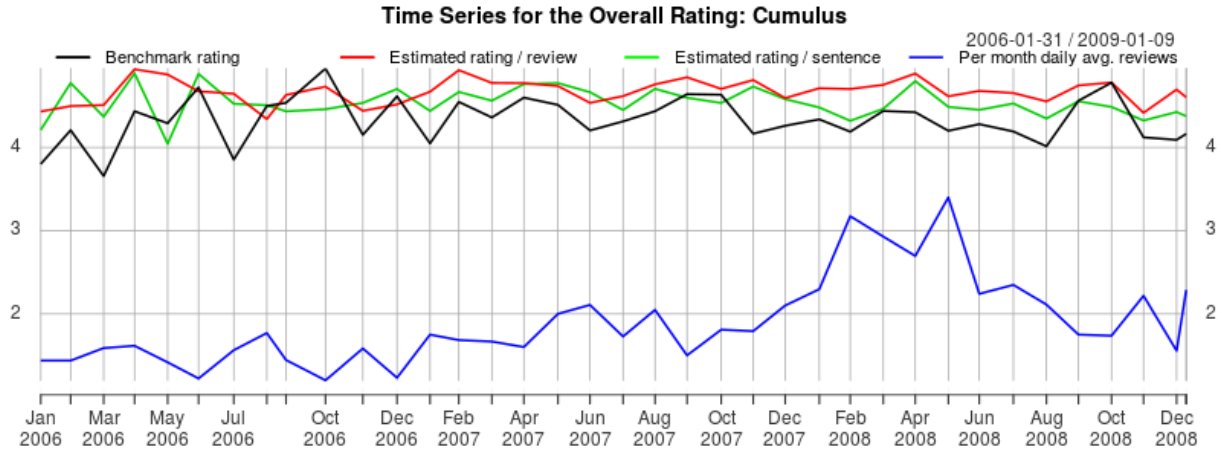

Figure 6: Time series for the *Overall* rating of the Cumulus hotel.

6.2 Value ratings

For the *Value* ratings of the "Noctilucent" hotel we notice that the estimated data roughly follow the benchmark data but they are overestimated. In the graph, the black line contains interruptions. These are due to the fact that there is not available information on that aspect in the considered period. (See Figure 7).

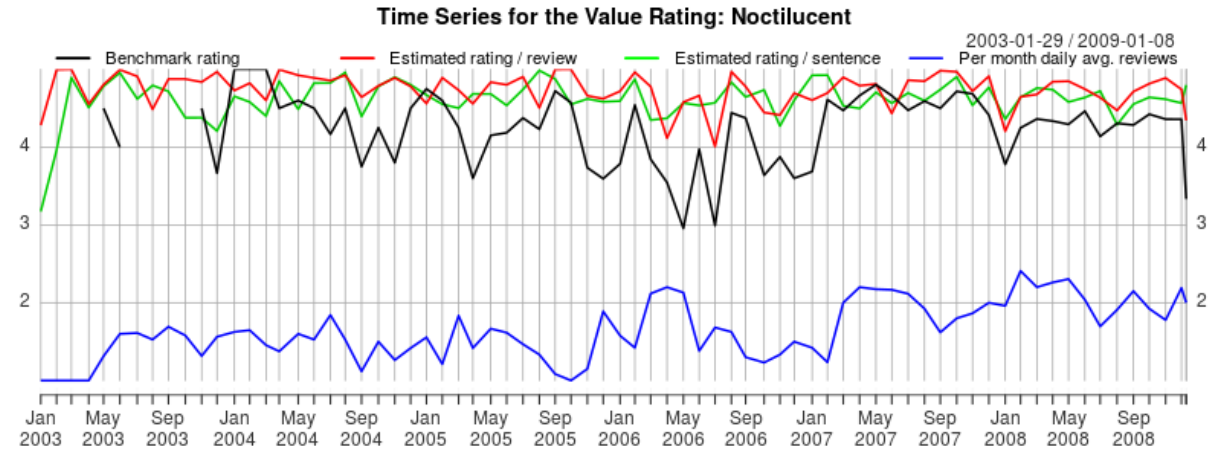

Figure 7: Time series for the *Value* rating of the Noctilucent hotel.

The time series trend of the estimated values for the "Cirrus" hotel is similar to the previous case. (See Figure 8).

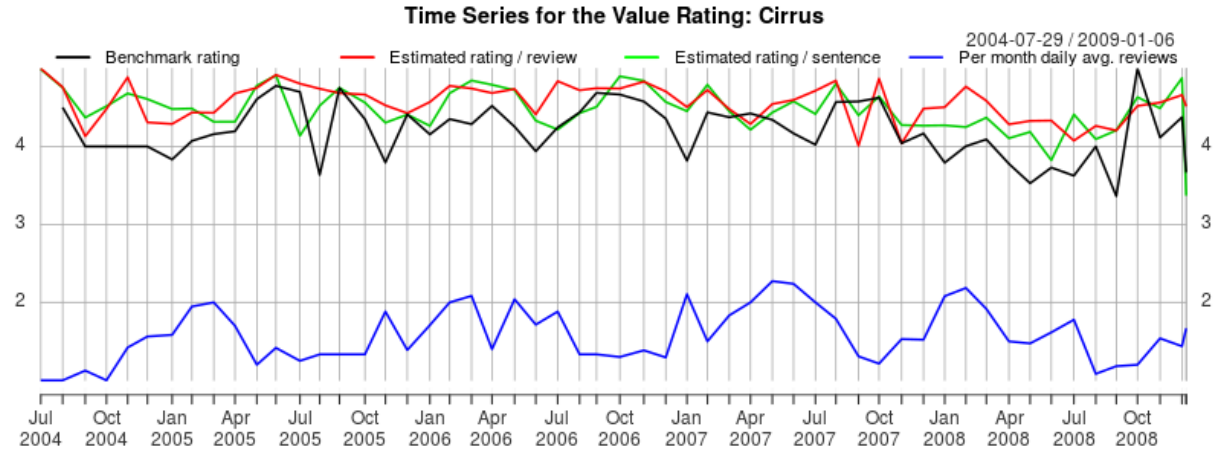

Figure 8: Time series for the *Value* rating of the Cirrus hotel.

The situation changes for the time series applied to the "Cumulus" hotel. The estimated values seem to be overestimated with respect to the benchmark data, even in the case where the average number of reviews increases. (See Figure 9).

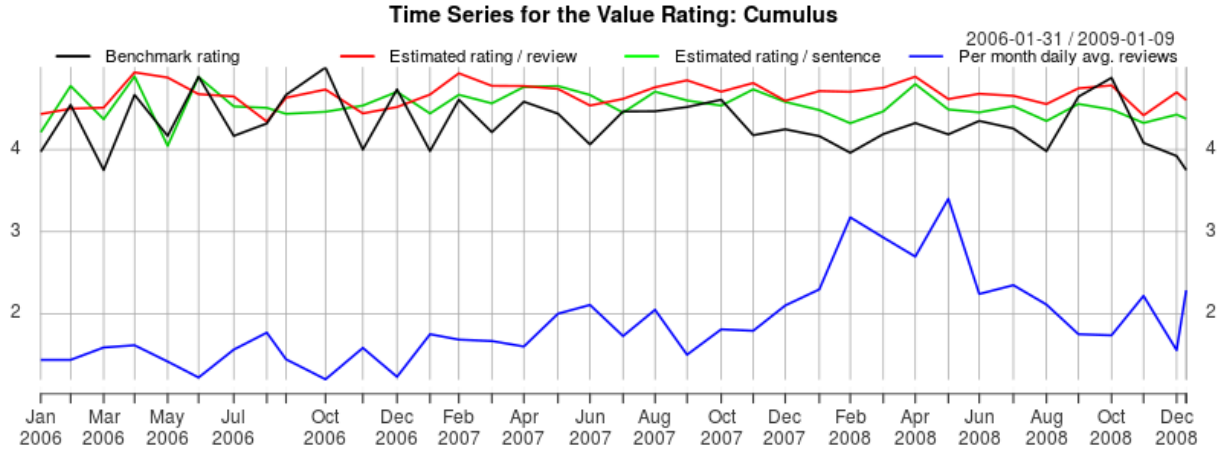

Figure 9: Time series for the *Value* rating of the Cumulus hotel.

6.3 Rooms ratings

For the *Rooms* ratings time series of the "Noctilucent" hotel we notice that `base` is sometimes overestimated sometimes underestimated while `sentence` is constantly underestimated except in June 2006 where all the values seem to overlap. Also in this case, there is missing data and it can be noticed in the interruption of the benchmark data line. The green line even if always

underestimating presents a pattern similar to the benchmark. (See Figure 10).

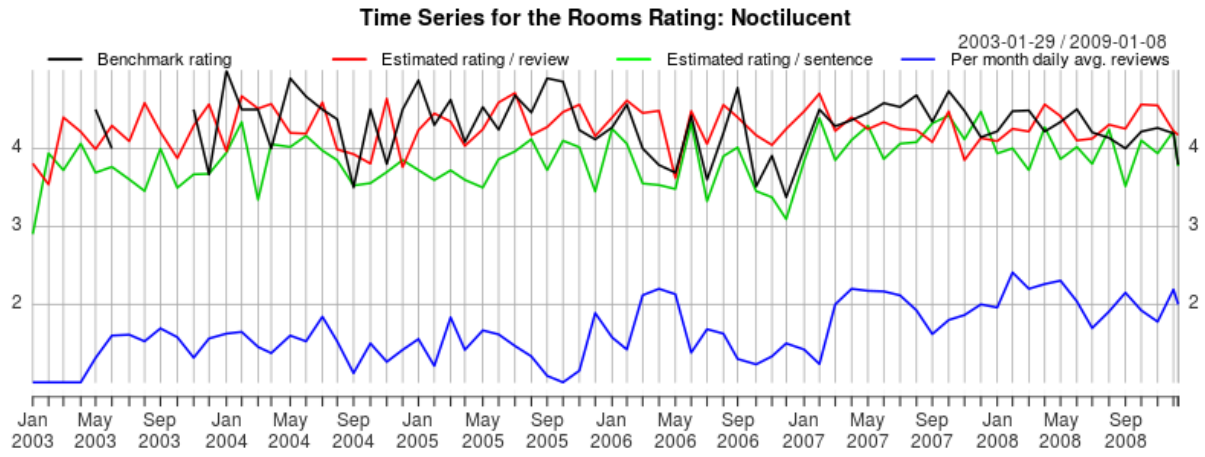

Figure 10: Time series for the *Rooms* rating of the Noctilucent hotel.

For the *Rooms* aspect ratings time series of the "Cirrus" hotel we notice that the `base` trend is mostly underestimated until April 2006 and then crisscrosses the benchmark data. The `sentence` trend is constantly underestimated independently of the number of the reviews. There is one single overlapping period in October 2008 where all the values overlap almost perfectly. (See Figure 11).

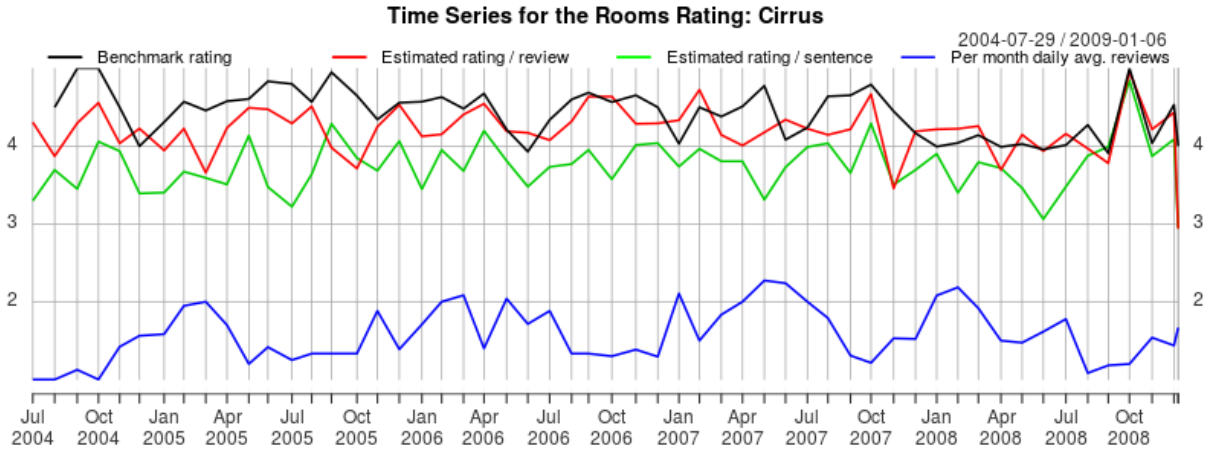

Figure 11: Time series for the *Rooms* rating of the Cirrus hotel.

For the "Cumulus" hotel for the *Rooms* time series, we notice that `base` follows a somewhat better trend than the `sentence` which is constantly underestimated. This situation occurs regardless of the increase of the mean number of reviews in a month. (See Figure 12).

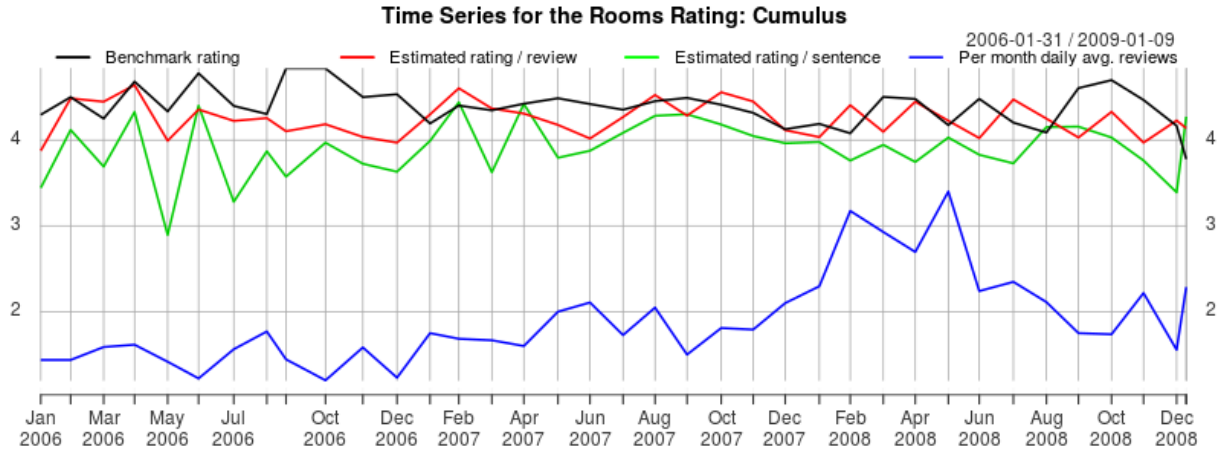

Figure 12: Time series for the *Rooms* rating of the Cumulus hotel.

6.4 Location ratings

For the *Location* ratings time series of the "Noctilucent" hotel we have a huge quantity of missing data. Both `base` and `sentence` have a similar trend in the absence of the missing data, roughly overlap between July 2006 and November 2006 then they are underestimated. (See Figure 13).

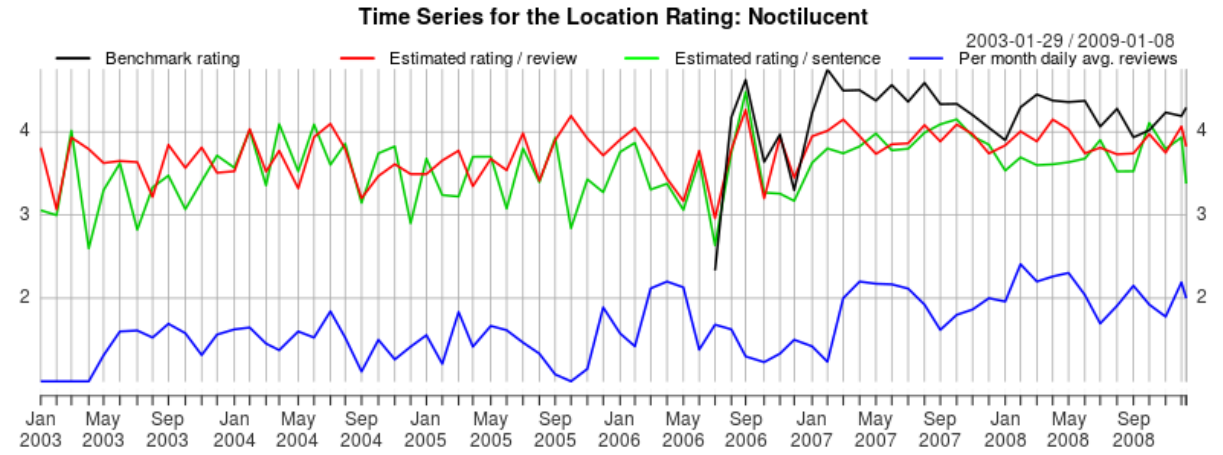

Figure 13: Time series for the *Location* rating of the Noctilucent hotel.

For the *Location* aspect ratings time series of the "Cirrus", there is around 50% of missing data. Compared to the existing benchmark data, both `base` and `sentence` are underestimated. The former seems to have lower overall estimated values than the latter. (See Figure 14).

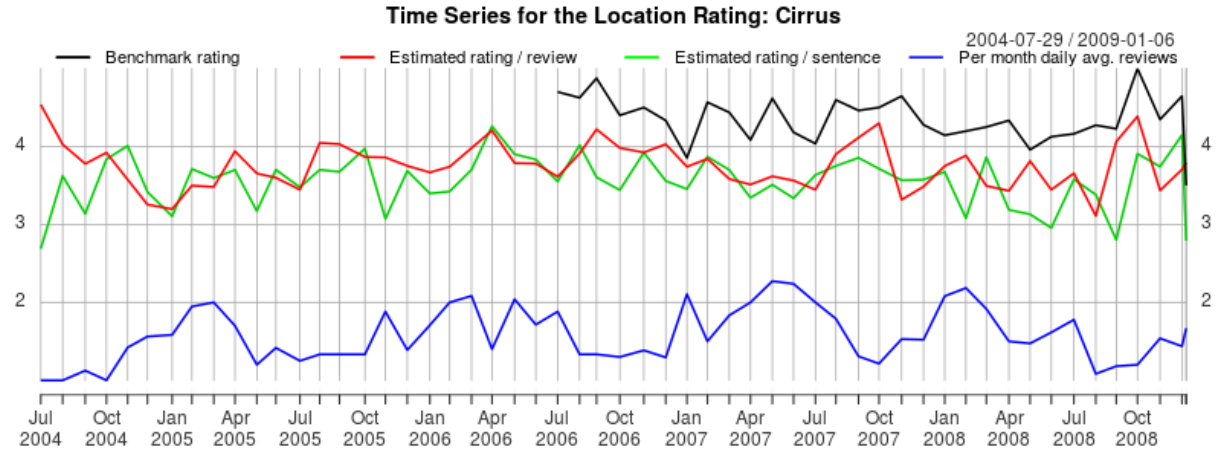

Figure 14: Time series for the *Location* rating of the Cirrus hotel.

The time series applied to the "Cumulus" for the *Location* aspect ratings have also missing benchmark data. Both `base` and `sentence` have a similar underestimated trend. (See Figure 15).

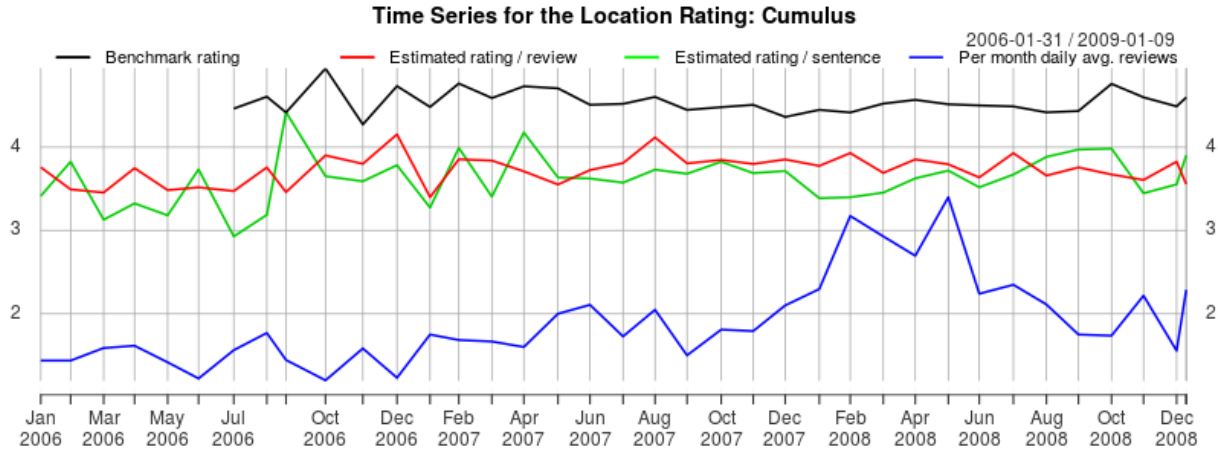

Figure 15: Time series for the *Location* rating of the Cumulus hotel.

6.5 Cleanliness ratings

The time series of the "Noctilucent" *Cleanliness* aspect ratings hotel has some missing data in the period until the December 2004. The `base` ratings seem to cross or overlap the benchmark data in may points. The `sentence` trend is similar to it but it presents underestimated values. (See Figure 16).

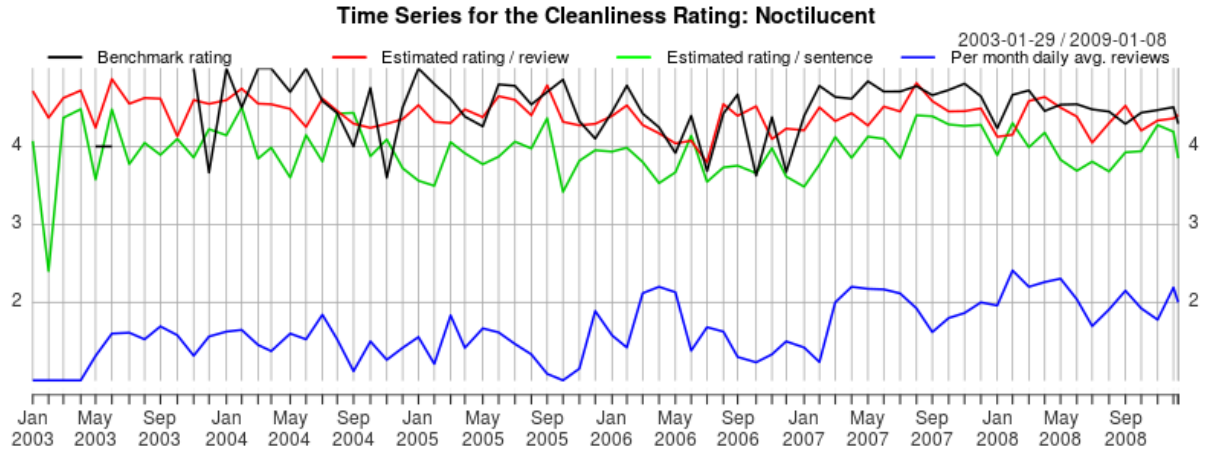

Figure 16: Time series for the *Cleanliness* rating of the Noctilucent hotel.

For the *Cleanliness* aspect ratings time series of the "Cirrus", we have the situation where the `base` estimates are roughly underestimated with respect to the benchmark data. The `sentence` estimated ratings seem to globally perform worse with highly underrated results present a few times as in January 2005, November 2006 and May 2008. (See Figure 17).

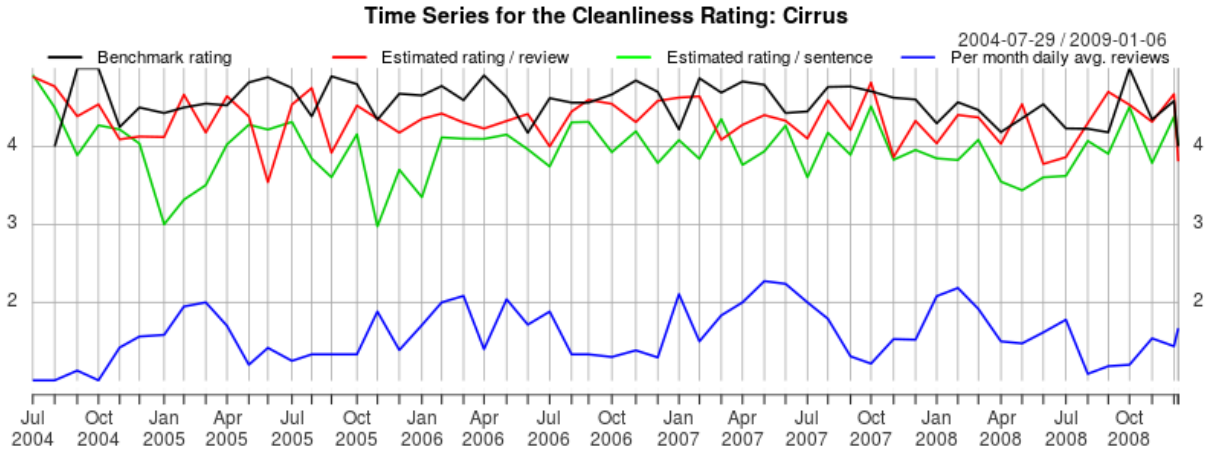

Figure 17: Time series for the *Value* rating of the Cirrus hotel.

In the case of the "Cumulus" for the *Cleanliness* aspect, we notice that the `base` has a similar trend as the benchmark data and frequently overlaps and crosses them. The `sentence` estimated ratings follow a more clearly discernible trend than the former but it is underestimated and in July 2006 the distance with the benchmark is deep. (See Figure 18).

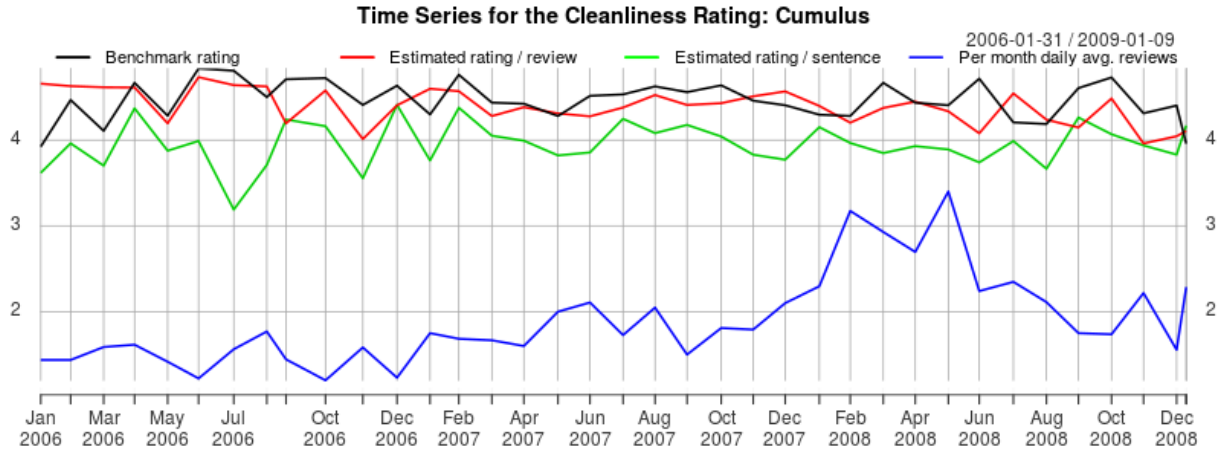

Figure 18: Time series for the *Cleanliness* rating of the Cumulus hotel.

6.6 Check In / Front Desk ratings

The time series for the *Check In - Front Desk* aspect of the "Noctilucent" hotel show a similar trend for `base` estimation ratings as well as for `sentence` estimation ratings. The difference is that the latter overlaps to the benchmark data while the former is quite distant. We have a lot of data missing though, benchmark information being available only for the time span between July 2006 and December 2008. (See Figure 19).

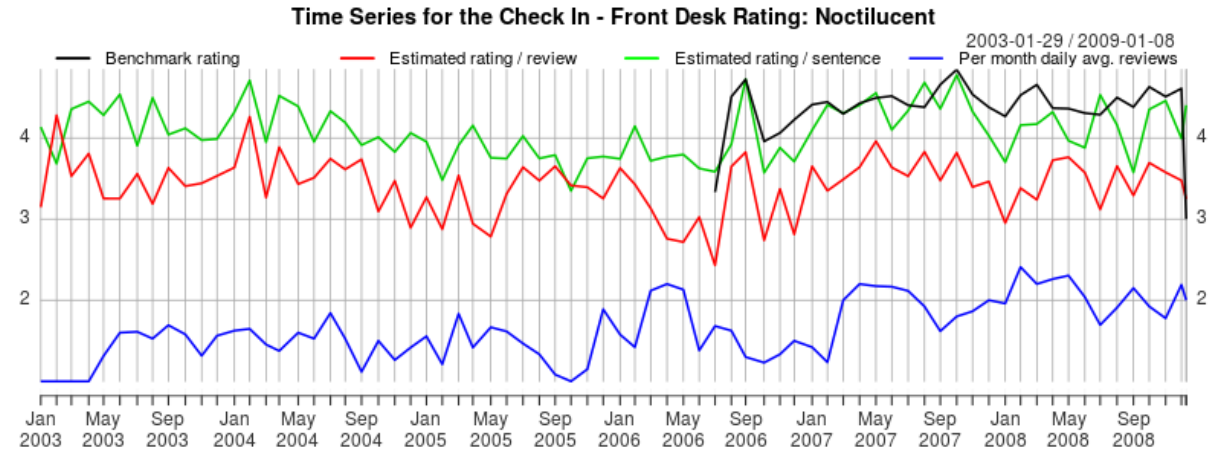

Figure 19: Time series for the *Check In - Front Desk* rating of the Noctilucent hotel.

For the *Check In - Front Desk* aspect ratings, the time series of the "Cirrus" has a similar behavior as in the previous hotel. Also in this case we have available data only in the interval

August 2006 - December 2008. (See Figure 20).

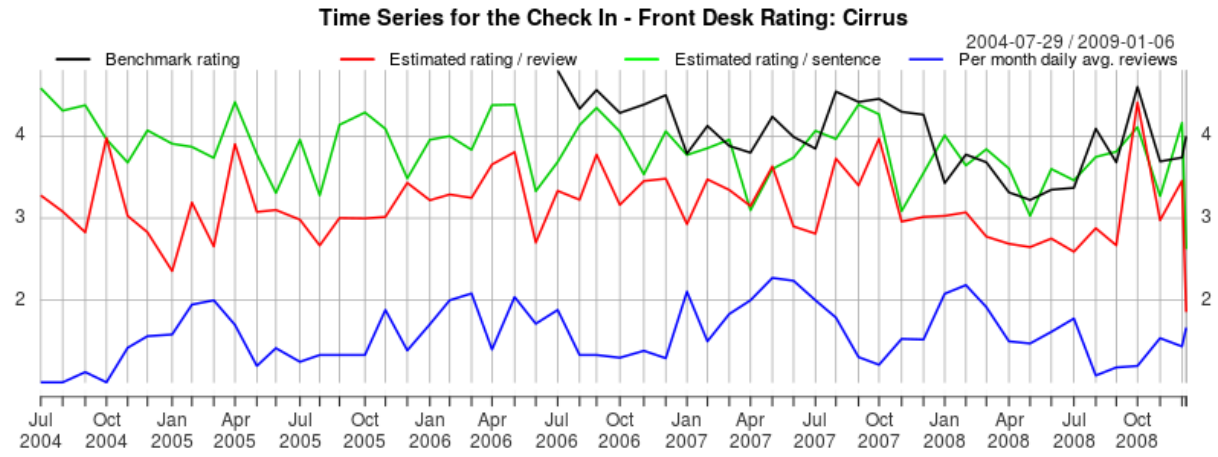

Figure 20: Time series for the *Check In - Front Desk* rating of the Cirrus hotel.

In the case of the "Cumulus" hotel for the *Check In - Front Desk* aspect, the time series behave as in the previous two cases, with the `sentence` estimated ratings having a good overlapping with the benchmark data while the `base` values are heavily underestimated. (See Figure 21).

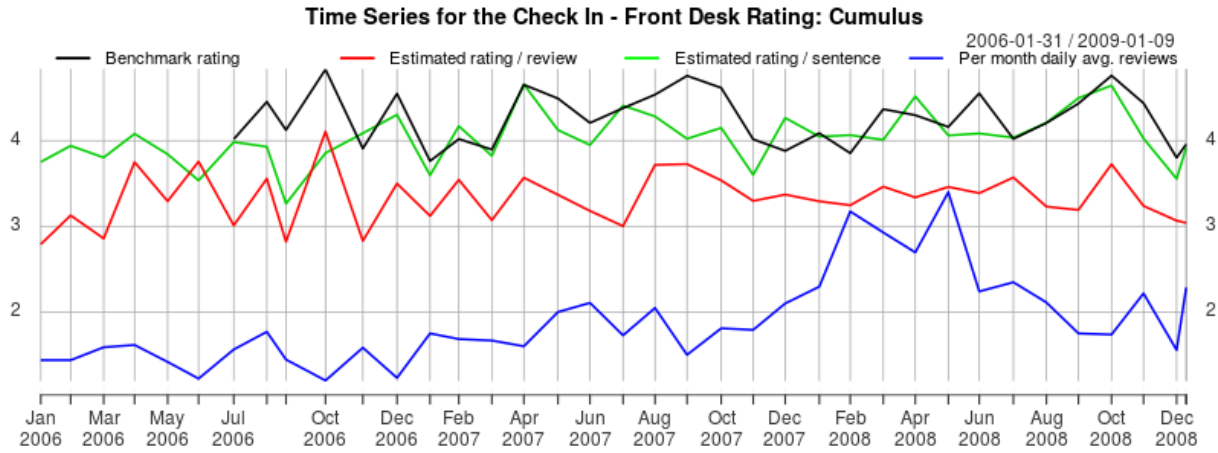

Figure 21: Time series for the *Check In - Front Desk* rating of the Cumulus hotel.

6.7 Service ratings

For the time series of the *Service* aspect ratings of the "Noctilucent" hotel we notice that the `base` estimated data is slightly overestimated while `sentence` data is underestimated. Also in this

aspect, there is a portion of missing benchmark data at the beginning of the series. (See Figure 22).

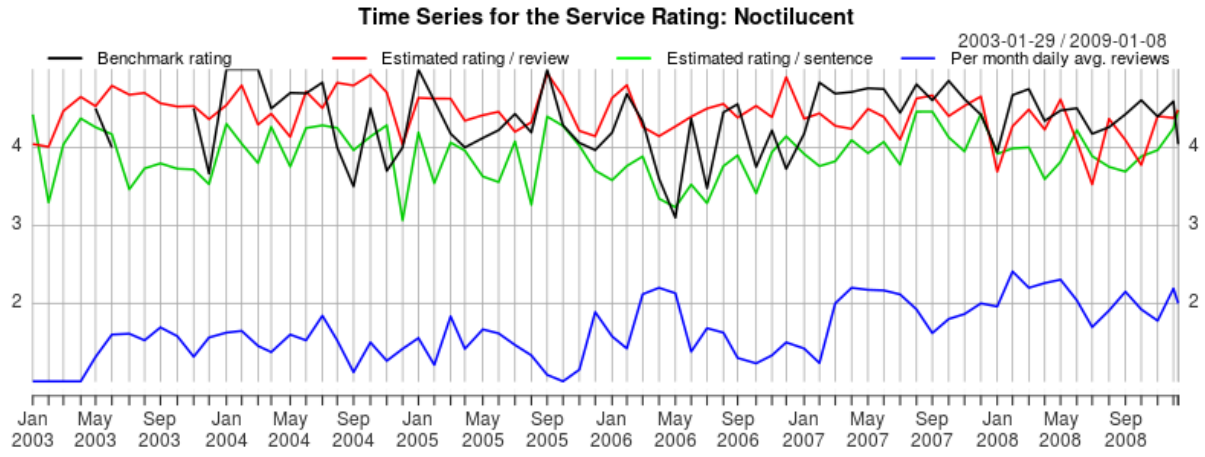

Figure 22: Time series for the *Service* rating of the Noctilucent hotel.

The time series considerations for the *Service* aspect ratings, of the "Cirrus" hotel are similar to the ones presented in the previous analysis. Both estimates follow roughly the benchmark trend with `base` estimated values somewhat overestimated and `sentence` estimated values slightly underestimated. There is a month of missing benchmark data for the present hotel. (See Figure 23).

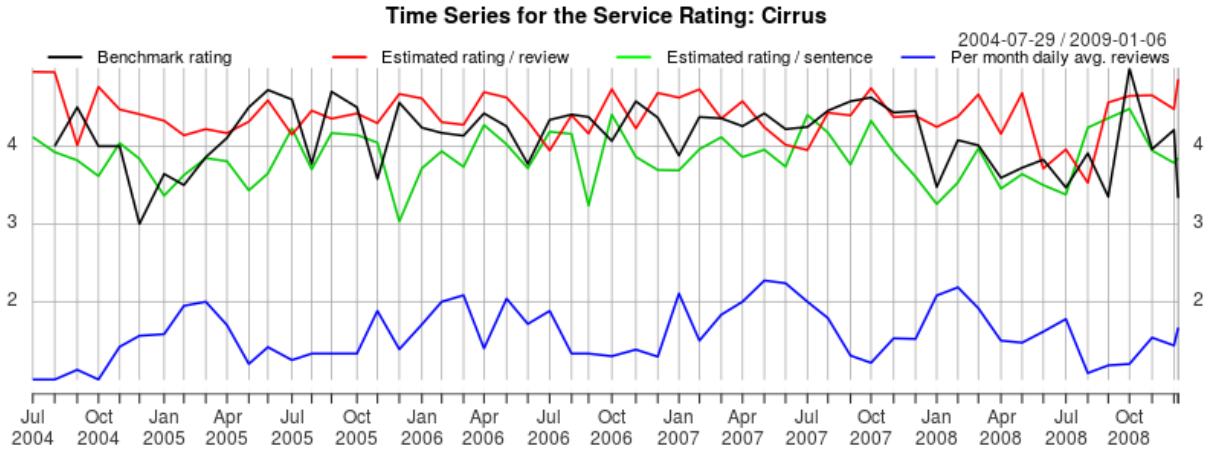

Figure 23: Time series for the *Service* rating of the Cirrus hotel.

In the case of the "Cumulus" hotel, there is a good trend for the *Service* aspect, although it is underestimated by the `sentence` estimated data that follow the benchmark with the exception

of the spring period in 2007. The `base` estimated data seem to be closer do the benchmark data but the trend is different. (See Figure 24).

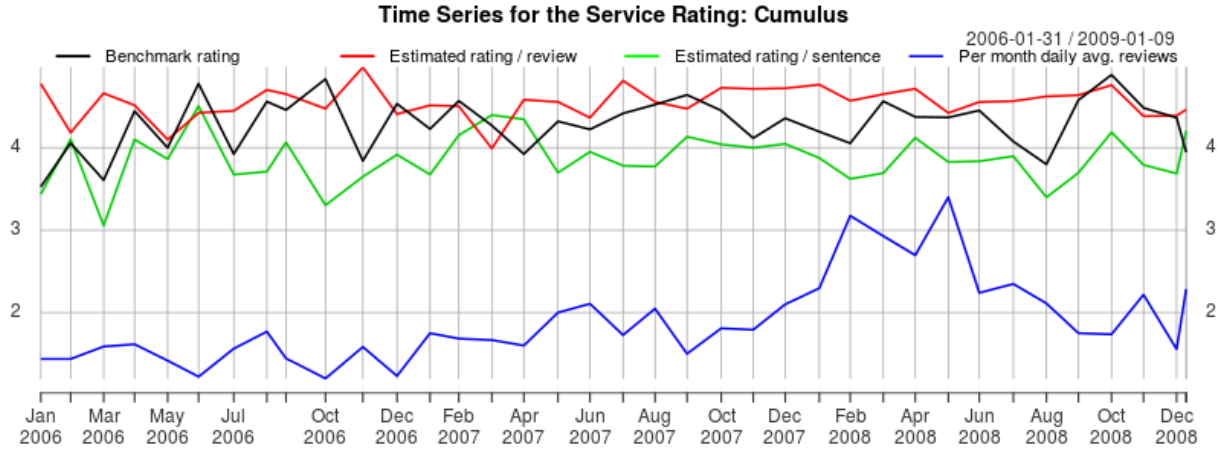

Figure 24: Time series for the *Service* rating of the Cumulus hotel.

6.8 Business Service ratings

The time series of the *Service* aspect ratings of the "Noctilucent" hotel have a reduced number of benchmark values. These are present from August 2006 until December 2008. Both `base` and `sentence` seem to follow the same trend for the applicable period but the latter seems to be closer and roughly overlapping to the benchmark data. (See Figure 25).

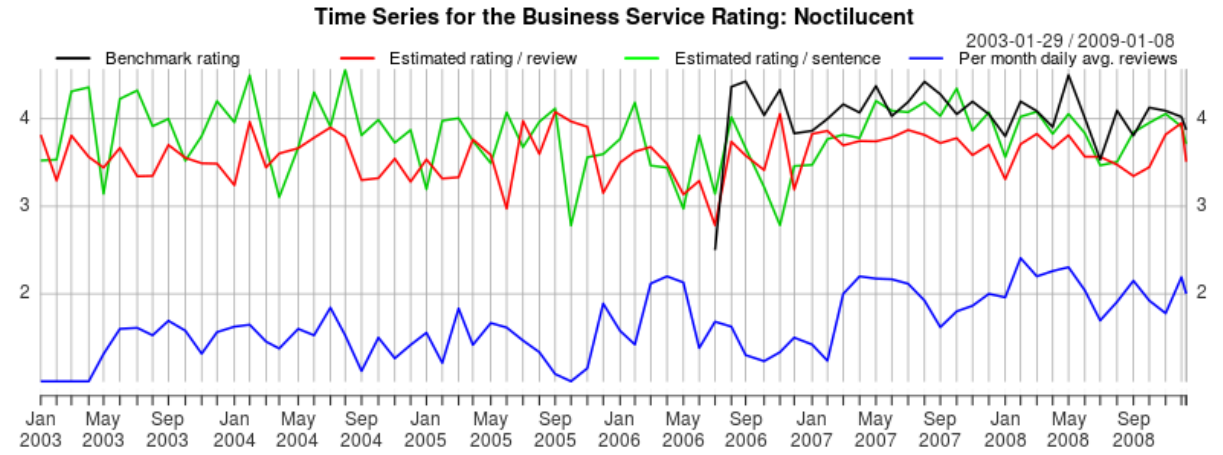

Figure 25: Time series for the *Business Service* rating of the Noctilucent hotel.

The time series considerations for the *Business Service* aspect ratings, of the "Cirrus" suffer

from the same lack of complete data as in the previous case. Nevertheless, estimated data seems to follow the benchmark data but it is unclear which of the `base` or `sentence` perform better. (See Figure 26).

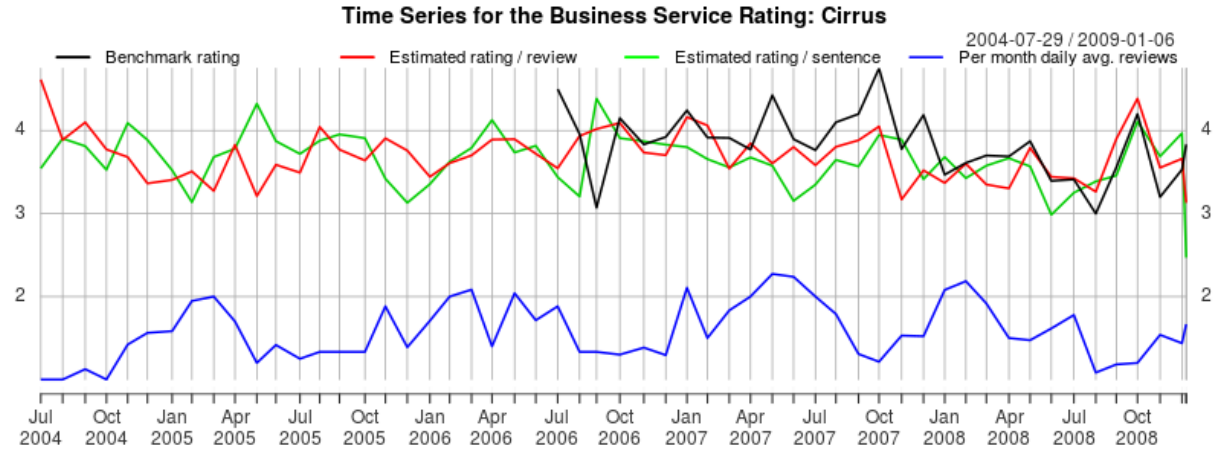

Figure 26: Time series for the *Business Service* rating of the Cirrus hotel.

The time series of the "Cumulus" hotel for the *Business Service* aspect has more benchmark data than the previous two cases. The `base` estimates seem to perform worse being roughly underestimated over almost all the time interval. The `sentence` estimates have a jittery trend between August 2006 and July 2007, after that it follows the benchmark data trend but slightly underestimated. (See Figure 27).

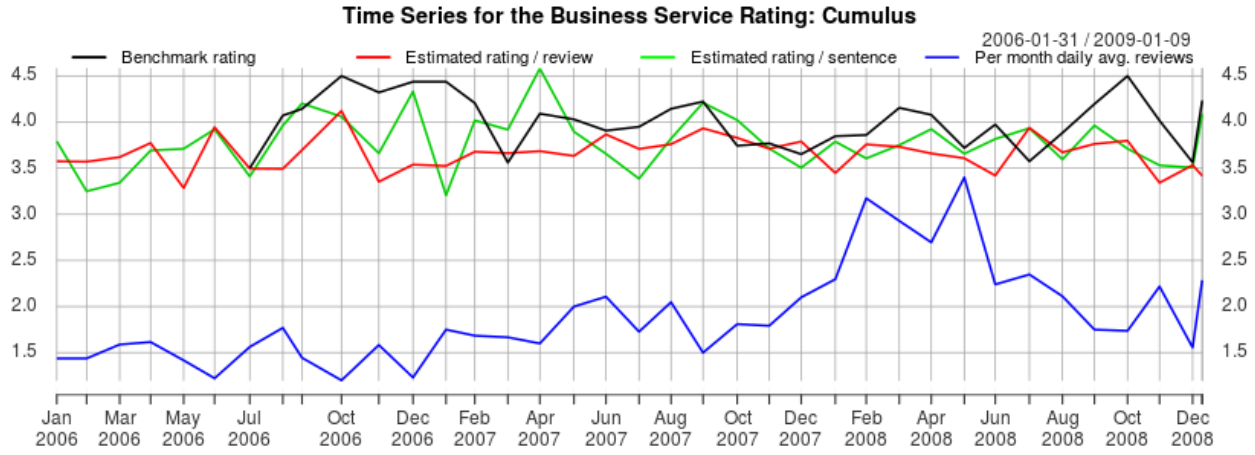

Figure 27: Time series for the *Business Service* rating of the Cumulus hotel.

6.9 Aspect weights

The aspect weights estimate the emphasis a person places on a specific aspect. The dataset is processed with the `sentence` version. We have picked up the four aspects that present the lowest number of NA values. In the case of the "Noctilucent" hotel, it seems that the reviewers put a lot of emphasis on the *Value* aspect and a lot less on the Rooms. The *Service* weight and the *Cleanliness* weight are situated somewhat in the middle of the previous two. (See Figure 28).

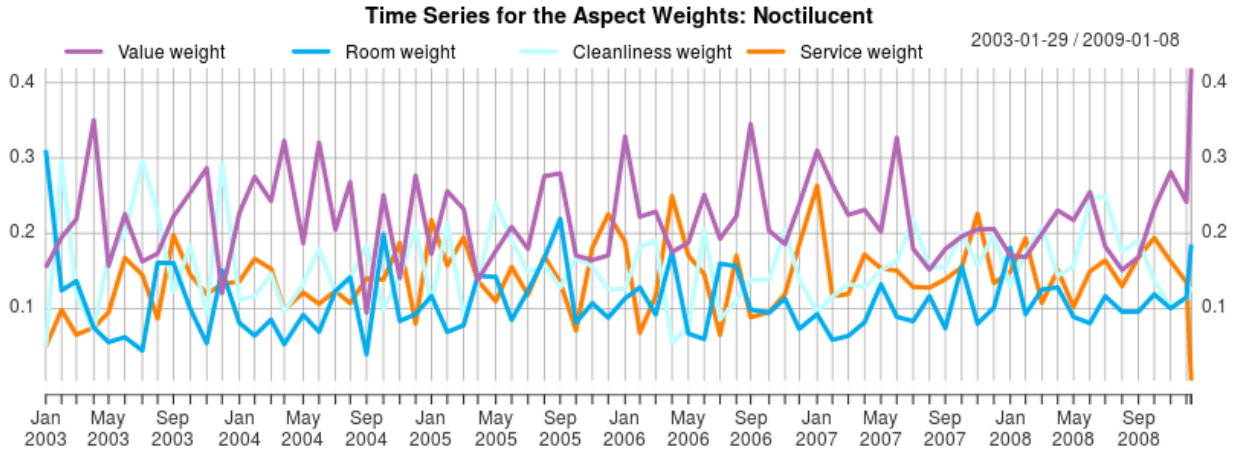

Figure 28: Time series for the *Value*, *Rooms*, *Cleanliness* and *Service* aspect weights estimation of the Noctilucent hotel.

The aspect weights for the "Cirrus" hotel seem to follow the same pattern, with *Value* being the most emphasized aspect and *Room* the least emphasized aspect. (See Figure 29).

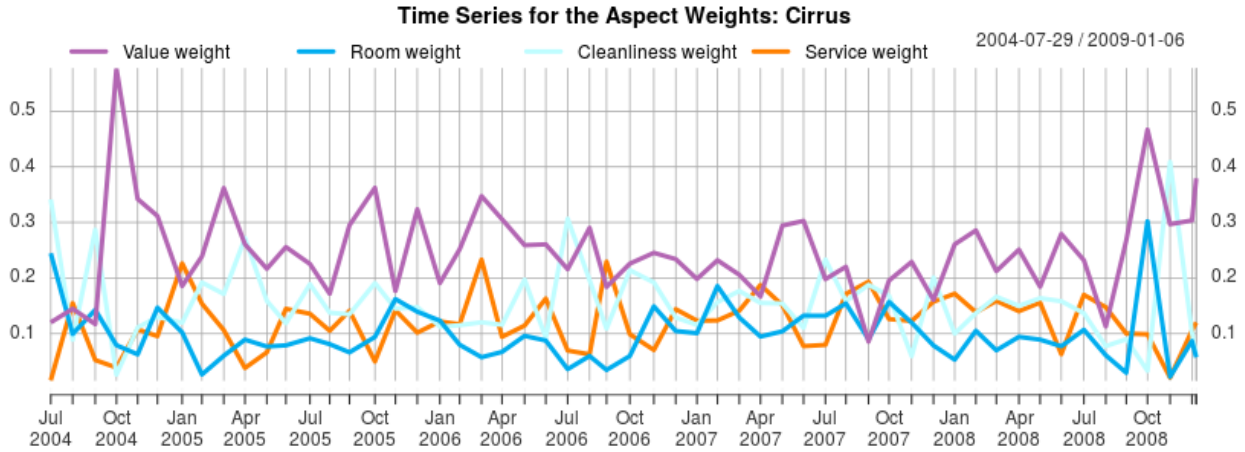

Figure 29: Time series for the *Value*, *Rooms*, *Cleanliness* and *Service* aspect weights estimation of the Cirrus hotel.

Similar pattern as in the previous case can be observed for the "Cumulus" hotel. (See Figure 30).

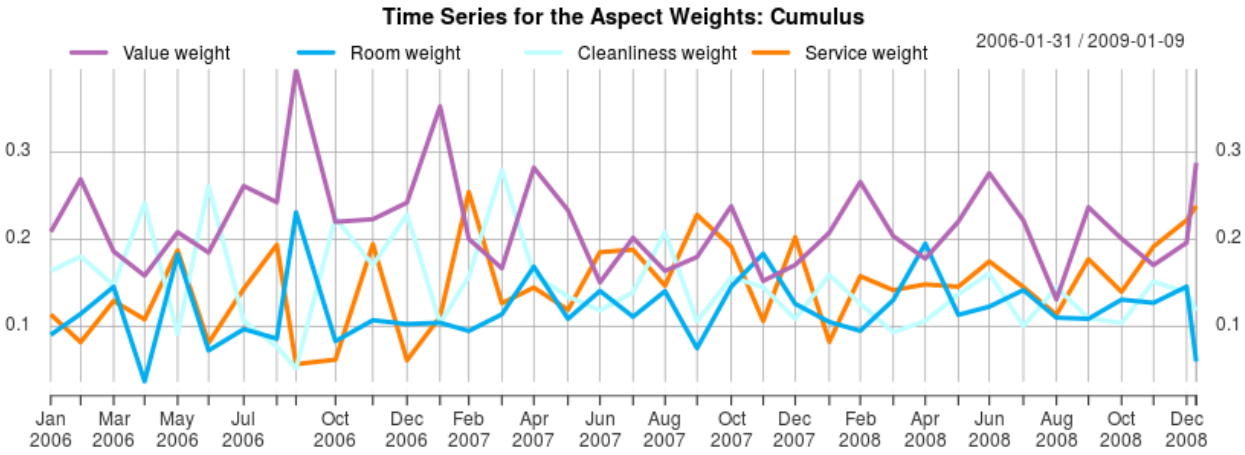

Figure 30: Time series for the *Value*, *Rooms*, *Cleanliness* and *Service* aspect weights estimation of the Cumulus hotel.

We surely cannot draw a conclusion just by considering 3 hotels out of 1118. While this approach might be acceptable for persons who follow the trends of a single hotel, we are interested in the comparison of the `base` and `sentence` estimation methods. We have thus computed the Root Mean Squared Error (RMSE) (e.g. Chai & Draxler, 2014) which is the average distance from the benchmark data to the prediction made by the model. It is computed for each of the predicted

values $\widehat{x_t}$ at time t of the benchmark value x_t for the number of the predictions considered, and it is given by the following expression:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(\hat{x}_t - x_t)^2}{n}}.$$

We have subtracted the RMSE values obtained with the two models from one another and plotted the results, see Figure 31. In this case, the values above the line indicate that the `sentence` algorithm performs better than the `base` whereas the values under the $y = 0$ line indicate that the `base` algorithm performs better than the `sentence`. From the boxplots we can conclude that, overall, the RMSE difference is favorable to the `sentence` model.

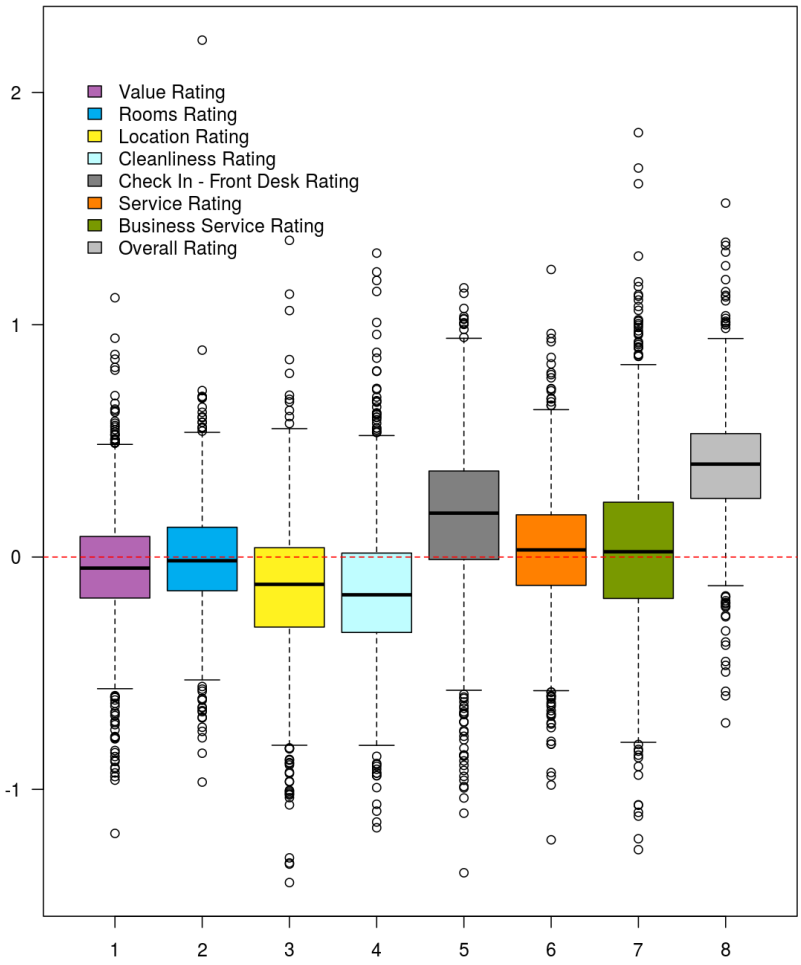

Figure 31: RMSE differences between `base` and `sentence` datasets for all the hotels.

7 Conclusions

The benchmark data limitations such as data incompleteness, the strong presence of the halo bias and the inappropriate number of aspects did not seem to be a strong obstacle to obtain meaningful results. The scatterplots in Figure 2 and 3 give, in both cases, a clear picture of the fact that benchmark and `sentence` or `base` data are positively correlated. The correlation coefficients, though, tend to be favorable to the `sentence` method. In order to propose an alternative comparison between the two methodologies we have applied a synthesis of the RMSE of the data predictions for each hotel. The results displayed in Figure 31 demonstrate that, overall, the `sentence` method yields better results. This is very clearly expressed in the *Overall* rating which has complete data, and, perhaps, gives a more accurate possibility of comparison between the estimated and the benchmark values. All in all both methods perform well but the results obtained tend to be more supportive for the `sentence` method.

Appendices

A Appendix A

```
<Overall Rating>4
<Avg. Price>$173
<URL>http://www.tripadvisor.com/ShowUserReviews-g60878-d72572-r23327047-
   ↪ Best_Western_Pioneer_Square_Hotel-Seattle_Washington.html

<Author>everywhereman2
<Content>Old seattle getaway This was Old World excellence at it's best.THIS is
   ↪ the place to stay at when visiting the historical area of Seattle. Your
   ↪ right on the water front near the ferry's and great sea food restraunts,
   ↪ and still with'in walking distance for great blues and jazz music. The
   ↪ staff for this hotel are excellent,they make you feel right at home. The
   ↪ breakfast was great.We did'nt have to travel far to have a good cup of JOE
```

```
   ↪  and a light meal to start our adventurous day off into one of the most
   ↪ beautifull city's in america. This hotel is in an area that makes it easy
   ↪ to get to any place you want to go and still find your way back, I highly
   ↪ recomend this hotel for your next visit to seattle.
<Date>Jan 6, 2009
<img src="http://cdn.tripadvisor.com/img2/new.gif" alt="New"/>
<No. Reader>-1
<No. Helpful>-1
<Overall>5
<Value>5
<Rooms>5
<Location>5
<Cleanliness>5
<Check in / front desk>5
<Service>5
<Business service>5

<Author>RW53
<Content>Location! Location? view from room of nearby freeway
<Date>Dec 26, 2008
<No. Reader>-1
<No. Helpful>-1
<Overall>3
<Value>4
<Rooms>3
<Location>2
<Cleanliness>4
<Check in / front desk>3
<Service>-1
<Business service>-1
```

```
<Author>KGBT
<Content>Wow, what charm! As a Travel Agent, I've stayed at quite a few hotel,
    ↪ but this is the only Historic hotel so far... I loved it! Had to go back
    ↪ for a personal stay. The decor is beautiful, the lobby furniture fits the
    ↪ time period is still comfy. The city view rooms are great - love the
    ↪ little balconies. Great breakfast, nice people, great location - The
    ↪ Seattle Underground Tours is a 1/2 block away. I've aready sent my folk
    ↪ there for a stay have told others.
<Date>Dec 14, 2008
<No. Reader>-1
<No. Helpful>-1
<Overall>5
<Value>4
<Rooms>5
<Location>5
<Cleanliness>5
<Check in / front desk>4
<Service>-1
<Business service>4
```

References

Beckwith, N. E., Kassarjian, H. H., & Lehmann, D. R. (1978). Halo effects in marketing research: Review and prognosis. *ACR North American Advances*, *5*, 465–467.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3), 1247–1250.

Free Software Foundation. bash (4.3.48(1) [unix shell program]. [Computer software manual]. (2017). https://www.gnu.org/software/bash/.

Hornik, K. (2017). StanfordCoreNLP: Stanford CoreNLP Annotation [Computer software manual]. (R package version 0.1-2)

Jadeja, N., & Pandya, A. (2014). Multi-aspect sentiment analysis with topic models modeling. *International Journal of Advance Engineering and Research Development*, *1*(5).

Jagarlamudi, J., Daumé III, H., & Udupa, R. (2012). Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 204–213).

R Core Team. (2017). R: A Language and Environment for Statistical Computing [Computer software manual]. Vienna, Austria. Retrieved from http://www.R-project.org/

Wang, H., Lu, Y., & Zhai, C. (2010). Latent aspect rating analysis on review text data: A rating regression approach. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 783–792.

Wang, H., Lu, Y., & Zhai, C. (2011). Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 618–626).

Conclusion

The first article of this thesis is an overview of the current techniques applied in text mining and analytics and presented a panoramic of the more frequently used methods. The second article introduces the method for discovering and estimating the aspects and the aspect ratings together with the aspect weights. The third article represents a practical application of the proposed method.

The method, although presents several points of strength, could be improved furthermore. In the first step, the aspects discovery through LDA could make use of a direct sentence-based approach. The topics discovered could be named perhaps using an automatically labeling method. In the second step, an ordinal regression model could be investigated as an alternative to the linear model employed here, though the required regularized estimation would require a careful study to obtain a satisfactory computational efficiency. Another improvement worth to mention is the use of a Bayesian methodology in which the discovery of the polarity of the words could be explored with a sentiment analysis method as a prior.

The proposed method applied to the Hospitality industry, namely to the reviews of the hotels and the correlated amenities offered, delivered encouraging results. Consequently, with the necessary modifications, it can be used in many other contexts. There is an abundance of companies that, through their website, gather important user-generated content regarding reviews of various products or services. An important source of user feedback can be found on the medical personnel on websites such as `HealthGrads`, `Doctor.com` and so on. Food service and Restaurant industry are also a common source of feedback found on `Amazon Restaurants`, `OpenTable` and many others. In the Apps world, applications are subjected to reviews too where `Apple Store` and `Google Play` are the most common stores. The Legal services receive feedback as well, not to mention the Automotive, Careers, Home Improvement or Financial industries, each of them having dedicated review services. There are companies that ambitiously gather information on a multitude of businesses, such as `Kudzu`, `Trustpilot`, `Yelp` just to name a few. They all collect reviews with an overall rating but they do not seem to segment the data into aspects. On such a broad application spectrum, the method proposed binded to some metadata can become an important decision-making tool for customers and managers alike.

Overall, the proposed method seems to be a reliable aid in discovering latent aspects of review data and estimating the related values. The findings of this work might be potentially useful for research fields such as Marketing or Management, and for tourism-related investigations.