Feature-based opinion mining: A proposal to enhance customer focus based on text feedback

UNIVERSITEIT VAN AMSTERDAM

Business Information Systems
MSC. INFORMATION STUDIES

Author: Antigoni Kourou
Supervisor: Dr. Maarten Marx

Student number: 11118172
Signature:

JUNE 2016

Dedication

Here goes the dedication.

Author's Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

Abstract

Online reviews are an important source for both customers, to obtaining information before making purchase decisions, and businesses to improve the quality of their services. Opinion mining is an active field of research that makes use of Natural Language Processing (NLP) techniques for mining the huge amount of online text reviews. This paper aims to enhance the focus on customers by analyzing their opinions and aligning the results with the quantitative feedback received. The setting of this research is the dataset of Airbnb reviews, a giant online marketplace for vacation rentals. Explicitly, this paper proposes the use of feature-based sentiment mining for estimating discrete score for each accommodation feature, based on online reviews. The proposed approach is composed by three stages: (i) feature extraction from full text reviews based on accommodation ontology; (ii) sentiment detection for each feature on sentence-level and hybrid-based method, including both rule-based and lexicon approaches; and (iii) data analysis of discrete sentiment scores per feature and overall. The aim of this proposal is to show how insights from text content can be aligned with the quantitative data of feedback systems for different analysis purposes. The two main analysis directions include firstly the comparison of ratings generated by text with the ratings of Airbnb system and its bias issue, and secondly the extraction of features and their corresponding sentiment based on text.

Table of Contents

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

The exponential growth of Information and Communication Technologies (ICTs) has played a great role in the development of tourism industry. Electronic tourism (e-tourism) is the application of ICTs for tourism purposes, including the digitalization of all its processes and value chains [**?**]. Increasingly, ICTs provide users with access to many sources of information and has eventually affected the consumer behavior in tourism industry [**?**]. In order to remain a strong competitor, service providers have to keep their customers happy. Nowadays there is a huge variety of web services which provide a great complexity and diversity of recommended offers for meeting the demands of travelers. However, the personalized consumption patterns and individualistic lifestyles make it difficult for the service providers to anticipate tourists' behavior [**?**]. For decision makers to successfully understand the requirements, needs, desires and preferences of the customer, detailed information has to be obtained. Travelers, making use of the ICT tools that facilitate the information retrieval and decision making processes, have now direct access to all types of information provided by tourism agencies, companies, marketers, enterprises or other users. Furthermore, ICTs and Internet have transformed e-tourism markets from customer-centric to customer-driven [**?**], meaning that users play a major role in creating and sharing traveling information through blogs and review websites. Online feedback mechanisms, also known as reputation systems, *have emerged as a viable mechanism for fostering cooperation among strangers in such settings* [**?**]. Examples of these systems, for instance TripAdvisor, Booking.com or AirBnb, after each trade encourage both parties to give feedback about their trading partner based on their own experience. Customer feedback is an essential component in every modern business' tool kit. A big number customer feedback software tools exists for helping service providers to measure and improve customer satisfaction, identify

unhappy customers, reduce churn and get valuable insight from customers. However, most of these tools are very expensive, considering that most of service providers can make use of their own feedback mechanisms.

1.1 Problem statement

The most common types of consumer generated feedback are ratings from 1-5 stars and general text comments. An important separation exists between the distinct role of text comments as tacit knowledge and ratings as explicit knowledge and the ways they are analyzed. For online marketplaces to succeed, their feedback technologies must be able to not only collect users feedback, but to properly analyze it and utilize for decision making purposes [**?**]. However, current online travel systems, aiming to assist the consumer in finding suitable offers, filter the information based on location-price factor and on the overall ratings accumulated from the feedback system, meaning that text reviews are revealed for the public to read but they do not directly affect the overall analysis. Focusing on solely numerical ratings and ignoring the importance of text feedback leads to two major issues for feedback systems.

First, many academic papers on online reputation systems and building trust in the online marketplace report the existence of bias in online reviews [**? ? ? ? ? ?**]. This bias is seen as a tendency of reviewers to systematically highly rate their experiences, thus reducing the difference between the ratings in the system and the true values is an important issue towards a more efficient online feedback system. These over-estimated high ratings create to the customers the idea of "perfection" in all directions, which doesn't leave room for discovering their possible negative features. Thus, utilizing only biased ratings produces many results that falsly clasify the listings as extremly good. On the other hand, the analysis of ratings does not indicate much information for the service providers, who aim to acquire knowledge on how to improve their services. In order to gain detailed insights from the feedback system, some service providers including Airbnb, ask its users to rate not only the overall quality of the listing, but also six accommodation features. However, since the overall ratings are biased [**?**], how do we make sure that the ratings for specific features are objective? From this point of view, the bias on quantitative data of the system raises the issue of reliability on the system itself.

Second, the incompleteness of the analysis based solely in quantitative data leads to the need for complementary analysis. Customers' needs are considered multi-dimensional and difficult to measure on discrete scales such as ratings [**?**], therefore a customer has to extract the needed information from different sources and types of information provided by agencies, companies or other users. According to [**?**], text comments are particularly interesting for the audience as a new trust-building means in online marketplaces by revealing hidden knowledge, which is often underestimated from their owners and cannot be described by negative/positive ratings. Furthermore, [**?**] suggests that text opinions influence the decision making process even when

the ratings are high. In the Airbnb feedback system a negative rating is followed by a text in 45% of the cases, which implies the great power of text analysis for discovering deeper insights for the listing [**?**]. Acknowledging the importance of text comments, some feedback systems often offer summaries to all text comments, which mostly consist on a bunch of most used words. However, this bag of words does not necessarily cover the features that a certain user is interested in, neither the features that need to be improved. The users or service providers still have to read all the text comments related to the feature, meaning that it still does not reduce much of the work. A survey by [**?**] asked the respondents to indicate how many feedback comments they examined before each online transaction. The result showed that 81% reported examining 25 comments (one webpage), 5% viewed 50 comments, 11% more than 50 ones, and only 3% did not examine any text comments. These findings reveal that despite the importance of text feedback to the users, it is difficult for them to access the meaning of numerous text comments [**?**]. Given this situation, the average human reader will have difficulties on identifying and extracting the relevant information from the opinions in them. Automated analysis systems are thus needed [**?**].

1.2 Research question

Natural language processing (NLP) enable computers to derive meaning from any human written input, including their opinions. The NLP methods for doing so fall into the category of sentiment mining methods, known also as *opinion mining*. Examples of their application include mainly the movie rating systems (Netflix, IMDb) and the product rating systems (eBay). However, the importance of extracting sentiment of features from online reviews, besides their overall positive/negative sentiment is often ignored in the literature. This research proposes the implementation of feature-based opinion mining methods from complementing the analysis of customer feedback in e-tourism and accommodation market. The proposed approach uses an ontology based approach combined with sentiment mining techniques for generating opinion scores for each accommodation feature mentioned in the reviews of a feedback system. This solution deals with the two issues mentioned above, bias of ratings and the need for complementary text analysis for feature extraction. From its point of view, both issues can be brought together as one, since text analysis is believed to contribute to a better rating systems by reducing its bias [**?**]. By estimating sentiment scores for the text reviews, the pipeline transforms the text data into discrete quantitative form, which can easily be analyzed for different purposes. Thus, this paper focuses on two main questions:

> *First, how can opinion mining methods be aligned with the quantitative data analysis in order to enhance focus on customer feedback and produce detailed analysis results?*
>
> *And secondly, how good can feature-based opinion mining estimate the quality of accommodation features in e-tourism feedback systems?*

By answering these two questions, the purpose of the paper is to offer to service providers a new reliable approach on enhancing focus on customers, based on users' generated content.

1.3 Research rationale and structure

Customer Focus Theory is one of the essential parts of Information Studies (IS) and it serves as a guidance for business on how to put focus on their customers, as their most valuable asset. From Figure 1.1 can be clearly seen that Customer Requirement, Information, Feedback and Relationships are the four key factors of a good focus on customer [**?**]. In alignment with this guidance, businesses try their best on gathering customers information and feedback in several ways starting online forms, surveys and interviews to social media, forums and blogs. However, the most important part of work comes exactly after having the feedback, explicitly on how it is analyzed, interpreted and utilized. Most of business pay huge amount of money on outsourcing the feedback analysis, human resources (like the Amazon Mechanical Turk) and expensive software. From this point of view, the proposed approach is an added value for both service providers and their customer for several reasons. First, an analytic tool taking care of text feedback and bringing it into the traditional quantitative type of data, saves both a lot of time and money. Second, the insights and analysis are personalized to the business purposes, meaning that the winner is the one who gets the requirements properly. Thirdly, transforming text feedback into quantitative data serves as a measurable data-drive target over time, where businesses can check the quality directions of their services.

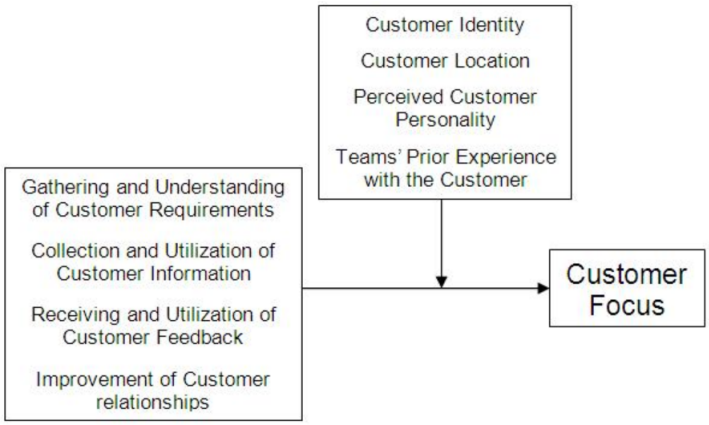


Figure 1.1: Customer Focus Theory. Source: [**?**]

From the literature point of view, it is noticed a gap in the alignment between text analysis with the analysis from quantitative data. This paper adds value firstly to the Customer Focus Theory and secondly to the use of feature-based opinion mining for analysis purposes. The existing systems and methods of feature-based opinion are examined and presented in the following chapter. Shortly, we can conclude that the literature lacks a system where all the

features belonging to a certain domain are analyzed and scored discretely based on text feedback. Furthermore, many reviewed papers[**? ? ? ? ? ?**] suggest the use of sentiment analysis for complementing the data analysis in feedback systems as an area in need for further research.

To provide an answer to the research questions, this paper is organized in seven chapters. The importance of gathering and analyzing customer feedback is introduced by the first part of the paper and it is followed, in the second chapter, by the current state-of-the-art of feature-based opinion mining used for analyzing customer feedback. The proposed approach is discussed in the third chapter by explaining each step of the pipeline developed for this research. The fourth chapter covers the methodology used in the research, from collecting the data to the analysis of the output. In chapter five, the whole proposed approach is evaluated in comparison to ratings given by humans. Afterwards the results are presented and visualized, which leads the discussion to the limitations of this work, its implications and further work to be done. Finally the paper concludes with a recap of the main findings.

CHAPTER 2

CURRENT STATE OF KNOWLEDGE

The task of extracting opinions from customer reviews is an issue, which has the attention of research communities for more than one and a half decade. With the continuous growth of Internet use, numerous social networks, blogs, news reports, forums, e-commercial websites and other platforms serve as an open-space for expressing opinions. The domain of interest varies from general public events to political campaigns as from product choices to marketing strategies. Sentiment analysis (SA) is the study of opinions, emotions and attitude towards an entity expressed in free text [**?**]. Sentiment analysis itself is a problem that includes several inter-related aspects, such as subjectivity classification, polarity determination, review usefulness measurement, opinion spam detection and so further. Due to the lack of standards in the methods used and the complexity of the problem, it is difficult to portray and categorize all the views presented in the literature. From 162 research articles covered in [**?**] during the time frame 2002-2015, my main focus are the feature-based opinion mining systems.

The reviewed systems base their extraction on statistical methods, unsupervised learning methods or ontologies. Statistical methods typically apply some rules on the frequency of nouns to find explicit features, unsupervised learning method group the explicit features and the similarities by measuring distributional properties of words and ontologies provide a pre-set list of features for the systems to be based. On the other side, there are mainly two approaches to polarity determination: machine learning and lexicon based methods. The level of sentiment analysis here differs from document level, sentence level or word level. The examined systems vary from each other on the methods used for feature identification, the granularity of sentiment analysis, the methods used for polarity detection or the way the output is presented. Due to the fact that this paper makes use of an ontology based approach, the examined systems are grouped

into non-ontology based and ontology based, although it is hard to compare the performance of these systems due to the lack of standard test data and methods.

Starting from non-ontology based approaches, Liu and Hu introduced in 2005 Opinion Observer, a lexicon-based system for sentiment analysis of opinions [**?** **?**]. Their system extracts product features by using association rule mining, explicitly the Classification Based on Association (CBA) algorithm. The system only uses adjectives as opinion words and assigns their polarity based on WordNet dictionary. The polarity of an opinion expression is determined in sentence level based on the number of positive/negative words. The final output are stored in a database in the form of *(feature, number of positive expressions, number of negative expressions)*.

In 2007 two other systems are proposed, namely Red Opal [**?**] and the famous OPINE [**?**]. Red Opal aims to score product features based on customer reviews. It uses probability-based heuristics for identifying frequent nouns and noun phrases for feature extraction and it is the only systems which makes use of review stars for defining the sentiment of features. In other words, the rating a reviewer would give are not calculated as an overall score but are assigned to specific features mentioned in their own reviews. A user interface is used for offering to the customers a ranked list of products based on the selected features.

The second approach, OPINE [**?**] is based on KnowItAll information extraction system which is domain-independent. This approach extracts explicit product features using the Pointwise Mutual Information (PMI) between phrases. It uses explicit features to identify potential opinion phrases based on assumption that "an opinion phrase associated with a product feature will occur in its vicinity on syntactic parse tree" [**?**]. After the extraction of the opinion expression, the semantic orientation of opinion words is determined by using relaxation labeling, an unsupervised classification technique.

Weakness Finder [**?**] is one of the well-known proposal of 2012. It aims to help the identification of features, and group these features into different aspects by using explicit and implicit features grouping methods, then judge the polarity of each sentence by using sentence-level sentiment analysis. The methods used for feature extraction are both collocation statistic based methods and lexicon-based on HowNet dictionary, thus a hybrid approach. The sentiment of features is detected in sentence level as positive, negative or neutral extending Liu's work, Opinion Observer system [**?**].

Bagheri et al. proposed in 2013 a novel unsupervised and domain-independent model for detecting explicit and implicit aspects in reviews for sentiment analysis [**?**]. In the model, first a generalized method is proposed to learn multi-word aspects and then a set of heuristic rules is employed to take into account the influence of an opinion word on detecting the aspect. Second a new metric based on PMI and aspect frequency is proposed to score aspects with a new bootstrapping iterative algorithm, works with an unsupervised seed set. Utilizing extracted polarity lexicon, the approach maps each opinion word in the lexicon to the set of pre-extracted explicit aspects with a co-occurrence metric. The output would look like in the example *(great -*

phone, price); (good - battery life, sound quality).

The first proper system using an ontology-based approach is proposed in 2012. Fuzzy domain ontology sentiment tree (FDOST) [**?**] aims to discover sentiment polarities over product features based on ontologies, which unites traditional natural language processing (NLP) techniques with sentiment analysis processes and Semantic Web technologies. The authors first construct the hierarchical relationships among product's attributes, product's attributes and corresponding sentiments through fuzzy sets. Then they extend the hierarchy of FDSOT, including the extraction of sentiment words, product features and the relations among features. Finally, the polarity weights of features are assigned making use of Double Propagation. The output of the FDOST are the features and the polarity weights, for example *{cheap, +1626.729}* reveals that there are 1626 consumers expressing positive opinion about the price.

In 2014 Penalver et al. proposed an another "novel" ontology based methodology for feature extraction [**?**]. Their system has four stages: a) feature identification based on ontology mechanism; b)polarity assignment to each feature based on SentiWordNet and the relative position in each user's opinion; and c) a new approach for opinion mining based on vector analysis. The resulting polarity value is an Euclidean vector with three coordinates (x, y, z) for each feature counting the number of positive, negative and neutral opinion respectively. The sum of these vectors will be the global polarity expressed by a user or for a certain feature. However the sentiment classification of all the opinions is based on document level, meaning that for each feature at least 100 sentences are needed to extract its sentiment.

Lastly, in 2015 is introduced Type-2 fuzzy ontology called T2FOBOMIE, a novel extraction and opinion mining system based [**?**]. The system reads the customers' full-text query for hotel search and reformulates it for extracting the user requirement into the format of a proper classical full-text search engine query. The proposed system retrieves targeted hotel reviews and extracts feature opinions from reviews using a fuzzy domain ontology. T2FOBOMIE uses a classical extensible markup language (XML)-based ontology, Protege OWL. The sentiment of features is identified in lexicon-based method, making use of SentiWordNet for detecting sentiment score and then grouping it into *positive, positive, neutral, negative, negative*. The results are presented with the help of an user interface which requires the customer to write his full-text query and the output of the system would be a list of hotels, the corresponding sentiment polarity and a link to the website.

Table 2.1 gives an overview of the comparison within these systems based on methods they use for feature extraction, sentiment detection, domain of usage, language and output format.

2.1 Implications of literature review

The examined approaches are mostly found for English or Chinese reviews. A big limitation of the systems based on statistical and unsupervised learning methods is that they ignore features with

SYSTEM	FEATURE EXTRACTION	SENTIMENT DETECTION	DOMAIN	OUTPUT	LANG
Opinion Observer (2005)	Association miner, CBA	Lexicon-based (WordNet)	Product	Feature, # of positive expression, # of negative expression	English
Red Opal (2007)	Probability-based heuristics	Assign star rating	Product	User interface of ranked results	English
OPINE (2007)	Unsupervised, Web PMI	Relaxation labeling	Ind.	List of sentiment sentences	English
Bagheri et al. (2013)	PMI, DBA	Bootstrapping	Product	A co-occurrence matrix: (feature; opinion words)	English
Weakness Finder (2012)	Collocation statistics	Lexicon-based (Hownet)	Product	List of features with negative sentiment	Chinese
FDSOT (2012)	Fuzzy set, Ontology-based	Double Propagation	Product	Set feature;polarity	Chinese
Penalver et al. (2014)	Ontology-based	Dictionary-based (SentiWordNet)	Movie	Euclidean vectors of polarity	English, Spanish
T2FOBOMIE (2015)	Ontology-based	Lexicon-based (SentiWordNet	Hotel	List of hotels with positive/negative polarity on feature	English
The proposed approach	Ontology-based & lexicon-based (WordNet)	Rule-based and lexicon based (VADER)	Hotel	Matrix of discrete scores per each sentence and features	English

Table 2.1: Comparison of examined systems on feature-based opinion mining

low frequency of occurrence. This fact is not anymore an issue for the ontology-based systems, however they vary on the methods used for building the ontology as there lacks any unified, standard, multi-domain ontology. The sources used for building the ontology of my proposed approach will be described in the Methodology chapter. Secondly, the examined systems detect the sentiment of features based on positive, negative and neutral polarity and not in discrete scores. Grouping the polarity in this form guarantees better scores in the accuracy of algorithms, however it does not leave room for further analysis with the data. In addition to this, the third limitation is the way of assigning sentiment to features. Depending on sentence or document level, an overview of the number of positive/negative opinions, the sentiment sentences itself or either their summaries, do not contribute in getting deeper insights from text, neither reduce the workload of the customers and service providers for analyzing the data. Thus, this paper proposes the use of estimated discrete scores of sentiment per feature in order to get deeper insights in customers opinion from free text and align this with the quantitative data retrieved from the feedback system. This proposal will be explained in the following chapter.

CHAPTER 3

METHODOLOGY

The proposed approach for an ontology-based feature extraction and opinion mining is composed by two important parts: pipeline coding and data analysis. For the first part, the code is written in Python 2.7. In addition to basic packages, I have made use of NLTK 3.2.1, py2neo 2.0.8, vaderSentiment, langdetect 1.0.1, WordNet 3.0 etc. All the codes belonging to the three first phases of the pipeline, excluding data analysis and visualization which will be treated later, can be found in Appendix II.

3.1 Dataset

Data from Airbnb listings in the Netherlands and UK has been collected over the past year by HAL24K[1]. Over a number of months in 2015 HAL24K recorded the active listings within the Netherlands and the UK and the associated accommodation descriptions and aggregate review scores of these listings. In March/April 2016 HAL24K revisited this known set of Airbnb listings and, for those that were still active, downloaded all of the text reviews that had been written by Airbnb users that had stayed at each listing up until the time of access. Approximately 1.5 million reviews were downloaded by HAL24K and stored within a Neo4j graph database maintaining the relationships between who had written the reviews, the reviews themselves and the accommodation about which the reviews were written. This corpus is filtered to be focused only in the data of Amsterdam for simplicity purposes and reduces the number of reviews in

[1] http://hal24k.com/

71.2K, consisting in 2 356 listing which each listing has on average 30.2 reviews. The same pipeline can run in the bigger corpus, with the help of a more powerful machine.

3.2 Accommodation ontology

In knowledge management and Semantic Web research areas, ontologies are considered essential in order to describe various concepts and relationships between them. For the accommodation domain and related domains a few ontologies have been proposed, which cover different aspects in this domain. However these ontologies are often found as sub-ontologies of the e-tourism domain. In this research, the ontology of accommodation, shown in Figure 3.1, is based on several sources. Firstly HONTOLOGY [**?**], which is brought in alignment with Accommodation in QALL-ME, Tourist Accommodationsin DBpedia.org and Lodging Business in Schema.org, was initially created by following different scenarios of booking an accommodation and processing reviews in Web services. Secondly ACCO Accommodation Ontology [**?**], which is an extension of GoodRelations ontology [**?**], is based on Owl Ontology Language (OWL) and is supported by Google and Yahoo for the e-commerce accommodation offers. Besides these ontologies, features mentioned in the Web based accommodation services of AirBnb.com, Booking.com and TripAdvisor.com are covered. However, this paper extracts features for only a few part of the ontology, explicitly six features of Airbnb system: *accuracy, check-in, cleanliness, communication, location* and *value*. For building a list of synonyms, hyponyms and hypernyms for these terms of ontology in order to increase the range of matches found in the text, the pipeline uses human-interaction. Thus, for every term the suitable definition to accommodation field is chosen manually in this case. The lists of related term is based on the Synsets and relations between concepts as introduced by WordNet 3.0 library and which is read by Python 2.7. They can be found in Appendix III.

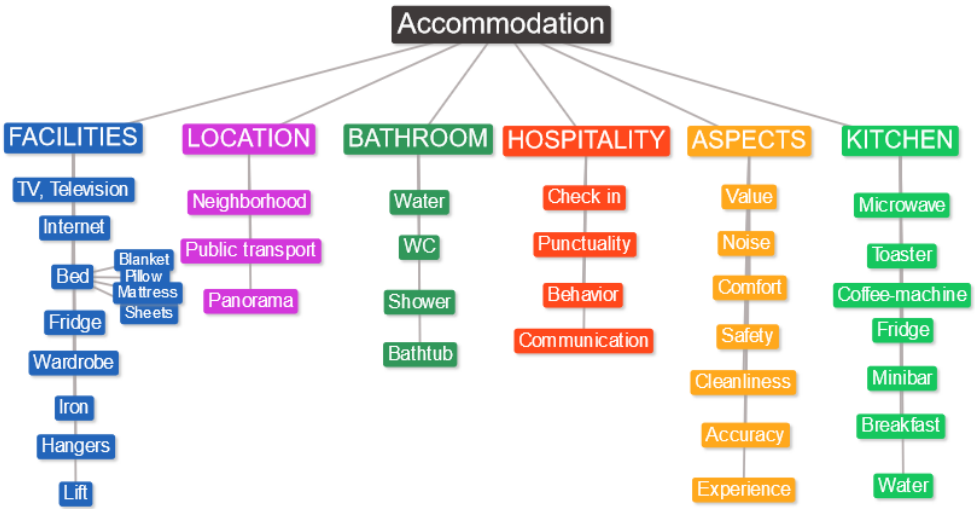


Figure 3.1: Example of the accommodation ontology

CHAPTER 4

THE PROPOSED APPROACH

For transforming the text feedback into quantitative data, which will serve as input for analysis purposes, this paper proposes an approach as shown in Figure 4.1. The pipeline consists of four main steps: *pre-processing, feature identification, sentiment detection* and *data analysis*. Initially, the whole input of the pipeline is the huge corpus of text reviews stored in Neo4J database. This big amount of data is considered to be very noisy, therefore it will be cleaned up as be described in the following section. Features and sentiment are then identified in text based in sentence level. The output of these three stages consists of the quantitative sentiment scores for sentences of reviews and the features identified in them. This output will then serve as the input data of the final stage of the pipeline: data analysis. The whole pipeline is built using Python programming language (Python 2.7.1) and its related packages. All the data analysis and visualization are written in Jupyter notebook 4.2. The following sections will explain each step of the proposed approach in further details.

4.1 Pre-processing

The pipeline reads the text data from cloud with the help of *py2neo 2.0.8*[1], a toolkit for working with Neo4J from within Python applications, and it formulates the queries in Cypher [**?**], the querying language for Neo4J graph database. The reviews in the corpus are read one by one and each of them is checked if it fulfills the language requirements. For this purpose, the pipeline use *langdetect 1.0.1*[2], a language detection library ported from Google's language-detection. In

[1] http://py2neo.org/2.0

[2] https://pypi.python.org/pypi/langdetect/1.0.1

my work, the whole concept of the pipeline is developed in English, as the most used language in e-tourism websites and as the most popular language in the research field of opinion mining. Thus, every time that the algorithm runs into a review not in English, it will ignore the review and continue to the next one. For each English review detected, the algorithm will cut the text into sentences, with the intention of detecting the sentiment in sentence-level and extracting

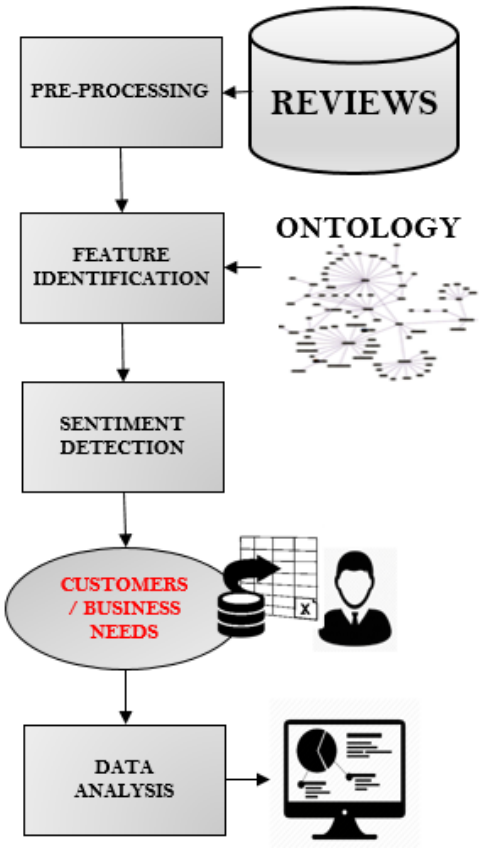


Figure 4.1: The proposed approach

features within these sentences. For splitting the reviews into sentences, the pipeline uses the NLTK Data package which is based in English punctuation marks. From this steps results that a review has on average 5.1 sentences, where the maximum number of sentences per review found is 41 sentences. The total number of sentences in English that form the dataset is 301 081 sentences. For all of them tokenization and lemmatization are performed to each sentence of the review. Tokenization is used form chopping the sentences into words, phrases, symbols and emoticons, each represented as a token. Lemmatization is the process of getting the root of the words, called lemmas, while ignoring its other forms as part of the speech. For tokenization the pipeline uses the *TweetTokenizer*[3] package, part of NLTK 3.0 Package and Twitter-aware designed, in order to be able to identify emoticons and adapt to new domains. On the other hand

[3]http://www.nltk.org/api/nltk.tokenize.html

lemmatization is performed using *WordNet 3.0*[4], a large lexical database for English terms and also part of NLTK 3.0 package. In WordNet nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept [**?**]. The output of the pre-processing steps is a list of lemmas for each sentence, which will be used by the pipeline for the next steps as explained in the following section.

4.2 Feature identification

As soon as each sentence in the corpus of reviews is represented as a list of lemmas, the ontology based approach is used to identify the accommodation features of each sentence. The lists of related terms of the features is lemmatized in the same way as explained above, and the duplicates are removed. The lemmatization process aims to create a list of related lemmas for each feature in the ontology, which will then be used for feature identification in the sentences. Thus, the feature identification step is a string match of the list of lemmas for every single sentence, with the list of related lemmas of every feature in the ontology. Considering that the ontology consists of many terms, this sample pipeline is trained to identify only six features that Airbnb asks the customers to manually rate as part of their feedback, namely *accuracy, cleanliness, check-in, communication, location* and *value*. However, the conclusion are drawn equally as all the features of ontology are included.

When a word in the sentence is identified to be a feature, the algorithm jumps to the next word of the sentence. Within one sentence more than one feature can be identified, as well as features can be mentioned more than once. Therefore, for every possible feature match, a *match counter* variable which keeps track of the features mentioned is assigned to the sentence. Here, the pipeline considers each sentence as independent and it ignores the logical connection between two sentences of the same review.

$$Feature_i = \begin{cases} k & feature \text{ mentioned k times} \\ 0 & feature \text{ not mentioned} \end{cases} \tag{4.1}$$

4.3 Sentiment detection

A very important part of the pipeline is sentiment detection. The algorithm used for this purpose is VADER (Valence Aware Dictionary for Sentiment Reasoning) [5], part of Python packages. VADER is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. For building the valence scores for sentiment intensity, VADER considers several well-known sources, such as ANEW, SentiWordNet 3.0 and SenticNet. These dictionaries

[4] https://wordnet.princeton.edu/
[5] https://pypi.python.org/pypi/vaderSentiment

include not only lexical features, but also grammatical and syntactical rules. Therefore, VADER is able to deal with negation, capital letters, degree modifiers, emoticons, punctuation types, contrastive conjunctions (*but, however*and slangs. Based on the comparisons of 22 sentiment mining tools of the last decade, VADER is ranked as the best algorithm for comments and the second best for social networks [**?**]. Some of its assigned scores are *(good 0.7); (great 0.9434); (awesome 0.8306); (dirty -0.83066); (terrible -0.9434)* [6]. In this paper, the pipeline uses VADER to return a sentiment score for every sentence of the review, which would afterwards serve as a discrete score for the identified features in that sentence. So, the algorithm returns a discrete score of sentiment per sentence, which we have to distribute to the features mentioned in it. Considering that in one sentence, more than 1 feature can be identified or the same feature can be identified more than once as mentioned above, a probabilistic model is developed. When calculating the sentiment we have four levels to deal with in hierarchical order: feature, sentence, review and listing. Starting from the top level, we have 2 356 listings, $L = \{ L_0, L_1, L_2 \dots L_{2355} \}$. Each listing in this corpus can be represented by the ID of the listing in the database *listing_id*. Every listing from this set has a number of reviews, which varies from one listing to the other and is identified from ID of the review $R_{listing} = \{ R_0, R_1, R_2 \dots R_r \}$. For a single review of a certain listing would be $R_{listing,\ review}$. This review is composed by a number of sentences $S_{listing,\ review} = \{S_0, S_1, S_2 \dots S_s\}$. In each of these sentences, the pipeline is trained to identify six accommodation features $F = \{accuracy, check\text{-}in, cleanliness, communication, location, value\}$, as explained in the previous section. When the sentiment of a sentence is detected, it does not particularly refer to a certain feature. Therefore, based on the sentiment of the sentence where the feature is mentioned, we have to calculate the feature's sentiment. 4.2 shows that the sentiment score (SS) of a feature in sentence level depends on the SS of the sentence itself, the number of times a feature is mentioned in this sentence and the total number of features mentioned in it.

$$SS_{(sentence,feature)} = k * \frac{SS_{sentence}}{n} \tag{4.2}$$

where k is the number of times the feature is mentioned in the sentence and n the total number of features mentioned in this sentence. Afterwards, calculating the sentiment in review or listing level is an easier task because it depends on the scores per sentence.

$$SS_{(review,feature)} \equiv^{def} \frac{\sum SS_{(sentence,feature)}}{s} \tag{4.3}$$

$$SS_{(listing,feature)} \equiv^{def} \frac{\sum SS_{(review,feature)}}{r} \tag{4.4}$$

for each *sentence* in *review* and for each *review* in *listing*, where s is the number of sentences in a review where feature is mentioned and r is the number of reviews in a listing that mention the

[6]The full lexicon can be found in: *https:github.comcjhuttovaderSentimenttreemasteradditional_resources*

feature. The calculation of sentiment scores of the features in sentence level is done as part of the sentiment detection phase of the pipeline and it is repeated for every single sentence. For clarifying its use lets take an example. Considering the reviews of a random listing, i.e. 24328, $R_{24328,\ 10146}$ would represent the review:

> *We had a great stay at Joe's. The handy guide to the house and neighborhood was much appreciated, as were Joe's clear instructions for checking in while he was away. The house was comfortable and eclectic, full of personal character.*

Thus, $S_{24328,\ 10146,\ 1}$ represents the second sentence of the text above. In this sentence can be identified three features *check-in, communication,* and *location* each once, thus for each of them k would be 1 and n would be 3. Figure 6.1 shows the results of the pipeline for this case. Then based on this output the sentiment of features in review and listing level is calculated in a later phase as part of data analysis.

SENTENCE	SENTIMENT	ACCURACY	CHECK-IN	CLEANLINESS	COMMUNICATION	LOCATION	VALUE
The handy guide to the house and neighborhood was much appreciated, as were Joe's clear instructions for checking in while he was away.	0.851896	0	0.28394	0	0.283936943	0.28394	0

Figure 4.2: Example of the probabilistic sentiment results

4.4 Data analysis

The last step of the pipeline and the most interesting one for service providers and businesses is data analysis. Up to here, we saw the pipeline processing all the text data step by step and transform it into meaningful sentiment scores for each sentence and each accommodation feature of the reviews. The output of these phases is a Comma Separated Vector (.csv) file, consisting of all listings, all listings' reviews, its respective sentences and the sentiment for the identified features on these sentences. This scores are analyzed with Pandas[7], an open source library providing high-performance, easy-to-use data structures and data analysis tools for the Python. The workspace of analysis and visualization of results is Jupyter Notebook, a web application that allows users to create and share documents that contain interactive code, equations, visualizations and explanatory text. For transforming the sentiment scores of reviews and listings into star ratings, the following coding scheme is used:

```
if (Sentiment < -0.75): stars = 1
        elif (-0.75 <= Sentiment < -0.5): stars = 1.5
        elif (-0.5 <= Sentiment < -0.25): stars = 2
                elif (-0.25 <= Sentiment < 0.0): stars = 2.5
                        elif (Sentiment==0.0): stars = 3
```

[7] http://pandas.pydata.org/

```
elif (0.0 < Sentiment < 0.25): stars = 3.5
        elif (0.25 <= Sentiment < 0.5): stars = 4
                elif (0.5 <= Sentiment < 0.75): stars = 4.5
                        elif (0.75 <= Sentiment): stars = 5
```

The data analysis in this step includes calculation and visualization of sentiment scores per reviews and listings, frequency of ontology-based features, ranking of listings based on overall sentiment or based on sentiment of one or more specific features, ranking of features for one listing based on their sentiment, identification of listings with the biggest number of reviews and identification on listings where the host have canceled once or more the reservation, computation and comparisons of sentiment scores from the pipeline with the ratings of Airbnb and so further. The analysis can of course be extended and customized to match the specific interest of the service providers or the customers. The main results of this analysis with be explored in the results section, followed by the implementation of pipeline according to its usefulness for service providers and customers respectively.

CHAPTER 5

EVALUATION

For checking the efficiency of the proposed pipeline, its performance is compared to ratings given by humans for two main purposes. First, the evaluation aims to find out how well VADER, the selected sentiment detector, performs in the chosen dataset and secondly, how well can the pipeline identify the features of a certain sentence of review. To answer these two questions, a sample of 100 sentences are randomly chosen from the dataset and are given to humans for rating. Each respondent is required to first read the sentences, then to estimate a score of sentiment for each of them on a scale from -1 to +1. In a second task, thes respondents are asked to mark which of the six accommodation features does the sentence refer to. These tasks are similar to what the pipeline is programmed to do. The evaluation form is completed by 5 humans, who come from different educational backgrounds, are geographically diverse and are Airbnb users. Their answers are analyzed and compared to the pipeline output using SPSS. Considering the diversity between users, their evaluations are firstly compared to each other using correlation values between samples and distribution statistics. This analysis showed that three of the respondents had very similar answers, consisting in a correlation varying between 72% to 83 % between each other and a very similar distribution curve. Based on these results, the chosen sample to represent the human ratings are the evaluations of the three "high quality" respondents.

5.1 Evaluation of VADER algorithm

To check the performance of VADER in the dataset, first the mean scores of the three respondents per each sentence are calculated. These mean scores represent the actual rating values, thus

VADER scores would be the predicted values. For measuring the differences between them, the chosen method is Root Mean Square Error (RMSE). The RMSE represents the sample standard deviation of the differences between predicted values and actual values. These individual differences are called prediction errors. The formula used for its estimation is:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{100} \left(y_{\text{i (Humans)}} - \hat{y}_{\text{i (VADER)}} \right)^2}{100}} \tag{5.1}$$

The RMSE between the two samples results to be 0.487, meaning that the VADER is not the perfect predictor of human ratings but considering that the Standard Error of Mean (SEM) in differences is 0.0489, we can conclude that there is no tendency of VADER to under-estimate or over-estimate the human ratings. From Figure 5.1 can be noticed that the two datasets share the same mean (4.125 compared to 4.13), the same median and mode (both equal to 4.5) and have similar standard deviations.

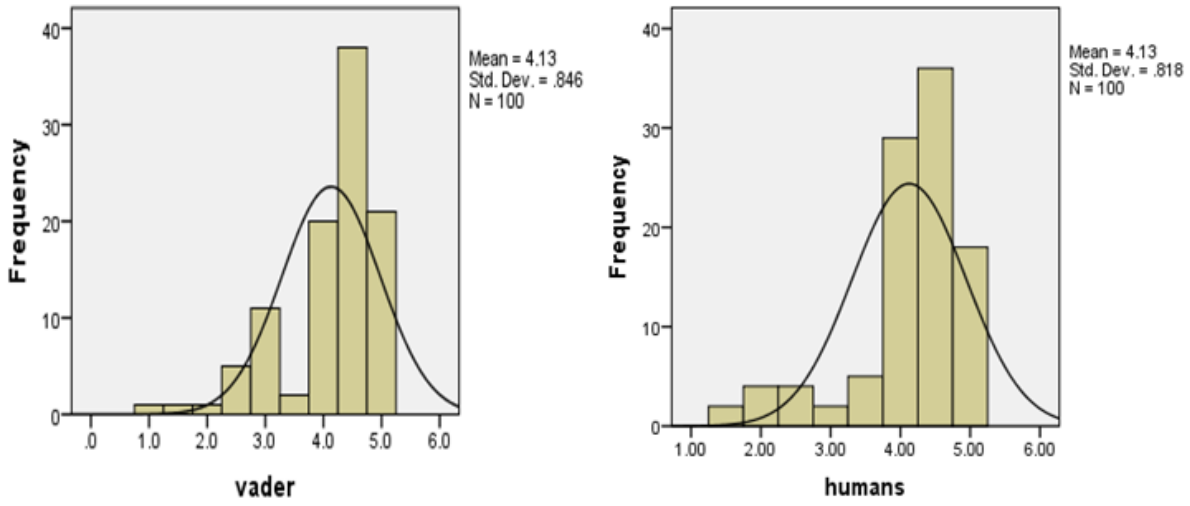


Figure 5.1: Distribution of VADER and humans' sentiment scores

The differences between VADER scores and human ratings result in 60% of cases when VADER has detected the exact score as human logic, 24% cases where the difference is just half a star, 14% cases with one star difference and the 2% cases with more than one star difference. The highest value of error is 1.5 stars, which means that VADER will never consider a very positive sentence as a very negative one and the other way round. However, it may consider these kind of sentences as neutrals or may be confused of the sign of sentences with slight sentiment.

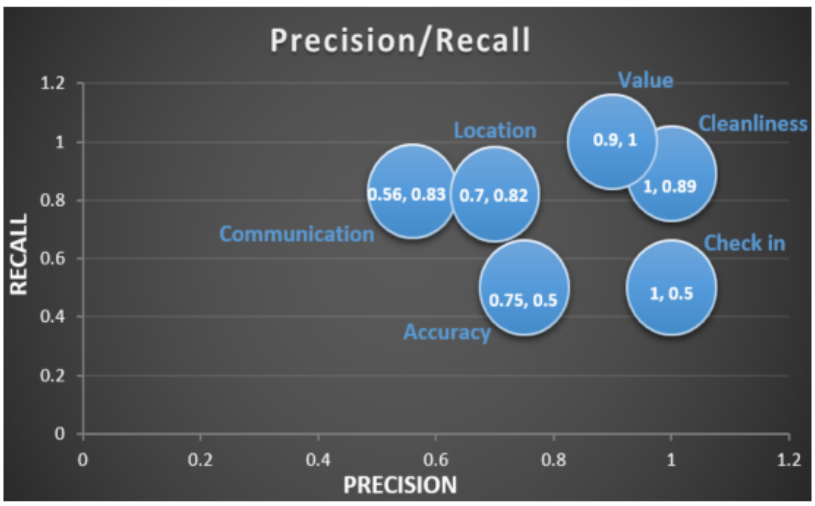
5.2 Evaluation of feature identification

The effectiveness of the proposed technique for feature identification is measured by using precision, recall and accuracy as suggested by [**?**]. The existence of a feature in a sentence is true when it is identified by the majority of respondents. Therefore for each feature, TP (true positives) is the number of sentences that the algorithm correctly identifies the feature, FP (false positives) is the number of sentences that the algorithm falsely identifies the feature; FN (false negatives) is the number of sentences that the algorithm fails to identify it and TN (true negatives) is the number of sentences that the algorithm correctly doesn't identify the feature. The formulas for calucating the precision, recall and accuracy of each feature are:

(5.2)

$$Precision(p) = \frac{TP}{TP+FP} \qquad Recall(r) = \frac{TP}{TP+FN} \qquad Accuracy(a) = \frac{TP+TN}{TP+TN+FP+FN}$$

In other words, *accuracy* refers to the quality of correct judgment of the algorithm about a feature. *Precision* refers to correctness in cases where feature is identified, and recall shows the portion of sentences that the pipeline fails to identify the feature. Ideally the algorithm shall have values of precision and recall close to 1 for each of the features. These values for each feature are illustrated in Figure 5.2, which shows that the algorithm manages to identify some features better than others. For example the *value* and *cleanliness* features of the listing are identified almost always with high precision and recall, but a low precision for *communication* means that the algorithm retrieves many FP. The opposite happens for *check in* when the precision is very high but many sentences fail to be identified. In average for the six features, the algorithm reaches the accuracy and precision 77.8% and recall 79.2%. Future work needs to be done in improving this step of the pipeline.



Feature	TP	FP	FN	TN
Accuracy	3	1	3	93
Check in	3	0	3	94
Cleanliness	8	0	1	91
Communication	5	4	1	90
Location	14	6	3	77
Value	9	1	0	90
OVERALL	42	12	11	35

Figure 5.2: Precision/Recall matrix for feature identification

CHAPTER 6

RESULTS

6.1 Overview of sentiment scores

Sentiment analysis and feature extraction from text reviews of Airbnb feedback system, provides us a set of discrete estimators for customers' opinions. From the analysis of reviews on Amsterdam dataset the most negative sentiment resulted to be -0.964284 and the highest score 0.998195. In total there are 12 798 negative sentences (sentiment lower than 0) and 236 904 positive ones, meaning the number of positive sentences is almost 18 times higher than the negative ones. Each review contains in average 5.1 sentences and a listing has in average 25.1 reviews. Figure 6.1a shows the sorted sentiment values, and it can be clearly seen that the slope of value [-1,0[is lower than in between]0,1]. Then Figure 6.1b shows the normalized frequencies of sentiment scores rounded by 1. From the graph can be noticed that the most frequent scores are 0, 0.6 and 0.8. The presence of so many sentences with 0 sentiment will be discussed later in this chapter. In addition, we can also notice a tendency of sentiment to avoid the "slightly" negative or positive zones around 0, which can have two possible explanations. First, the algorithm may fail in identifying slight sentiment and second, reviewers use text to express mostly strong sentiments. The sentiment scores are also grouped by accommodation features and for each of them its seen the distribution of these scores, the sorted patterns of values and the normalized frequencies. By comparing these distributions to each other and to the overall sentiment scores, it is noticed that the more the feature is mentioned the more similar the distribution is with the overall sentiment pattern. By the comparison we can also notice the sentiment scores most frequent per feature, for example for *value* would be *0.3-0.4*, for *location* is *0.6;0.8* and so on. This part of analysis and visualizations can all be found in Appendix I.

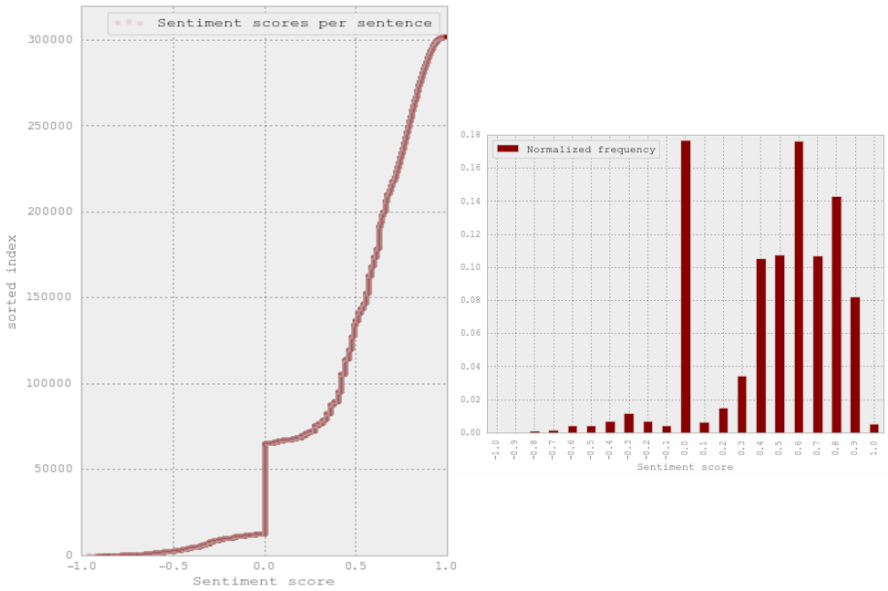


Figure 6.1: Sentiment scores of each sentence of reviews in the dataset

6.2 The case of sentences with "no sentiment"

As it is notices in for graphs of Figure 6.1 many sentences have sentiment score equal to 0, meaning neutral sentiment or no-sentiment. From the analysis results 52379 sentences have no-sentiment, consisting in 17.3% of the total sentences. From the analysis results that a part of these sentences are cases where an automatic cancellation. Every time that the reservation is canceled an automatic message is posted in the text feedback space as a review by the guest-to-be. Therefore without the cases of cancellation, in 16.09 % of the sentences the pipeline is unable to detect its sentiment, therefore it is seen as neutral. Furthermore from the analysis resulted 1259 cases of cancellations. Their occurrence is checked for each listings and from the analysis we could retrieve the listings which have the highest probability of canceling. It was noticed that in a rare case the number of cancellations reached 27, however the mean number of cancellation would be 1.7 per listing. In total 32.7% of the listings have canceled at least once the reservation, meaning that the customers should have to pay attention to the number of cancellations in one listing for making a safe choice. The cases of sentences with sentiment zero are excluded from the following analysis, in order to not affect the average sentiment scores per review or per listing.

6.3 Most mentioned features

An interesting point of view is the analysis of what features are most mentioned in proportion with the number of comments in the listing. This analysis is done in three levels: sentence level, review level and listing level. For all the three levels, we can clearly see that *location* is the most

mentioned feature by reviewers of Airbnb and *check-in* is the least mentioned one. From this comparison we can also indicate that the frequency of mentioning features is more accurate in review level. When considering the features mentioned in review level we make sure that the opinions of each reviewer are treated equally. Thus, no matter how many times the reviewer mentions the same feature within the review, his/her opinion will generate only one average sentiment score for it. In addition, in the sentence level the total number of sentences is very high which under-estimates the percentage. Similarly, in listing level it would take at least one sentence to indicate that the feature is mentioned in the listing, no matter if at one listing it is mentioned 100 times and at another just once, which would cause over-estimation.

	Sentences	**Reviews**	**Listings**
Accuracy	8364	7915	1773
Check-in	5818	5454	1623
Cleanliness	18440	17757	2076
Communication	16894	14610	2067
Location	69616	44539	2331
Value	19862	18811	2145

Table 6.1: Number of sentences, reviews and listings where features are mentioned

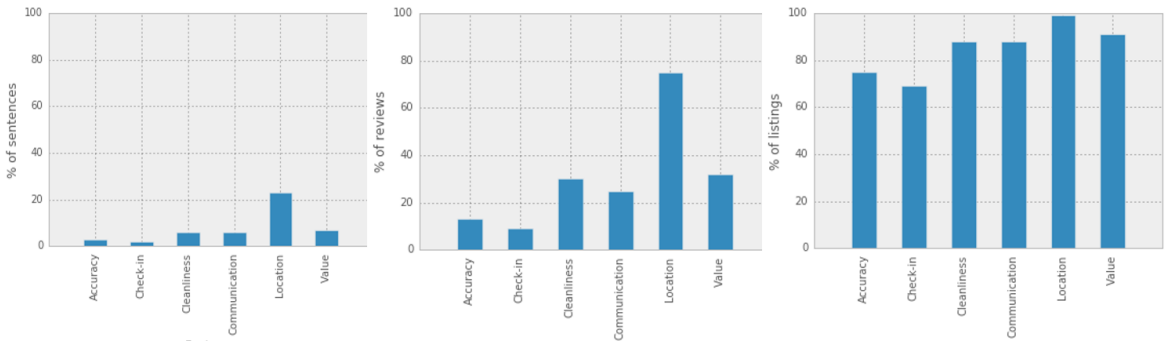


Figure 6.2: Percentage of sentences, reviews and listings where the features are mentioned

6.4 The trade-off between sentiment and number of reviews

Based on sentiment scores per reviews, Figure 6.3 represents for each listing the number of reviews where the feature is mentioned and it compares this value to the total number of reviews. In this way, we (as a customer) would prefer to choose listings which have many reviewers talking about a certain feature i.e. *location*, because it would make the sentiment score more reliable. For each listing the compound sentiment scores of explicit features are easily calculated. The second part of Figure 6.3 shows for the same ten random listings the compound sentiment scores of each feature and the overall sentiment of the listing based in all its reviews. In this way, the sentiment scores can be compared with the number of reviews. From Figure 6.3 we can see that for the

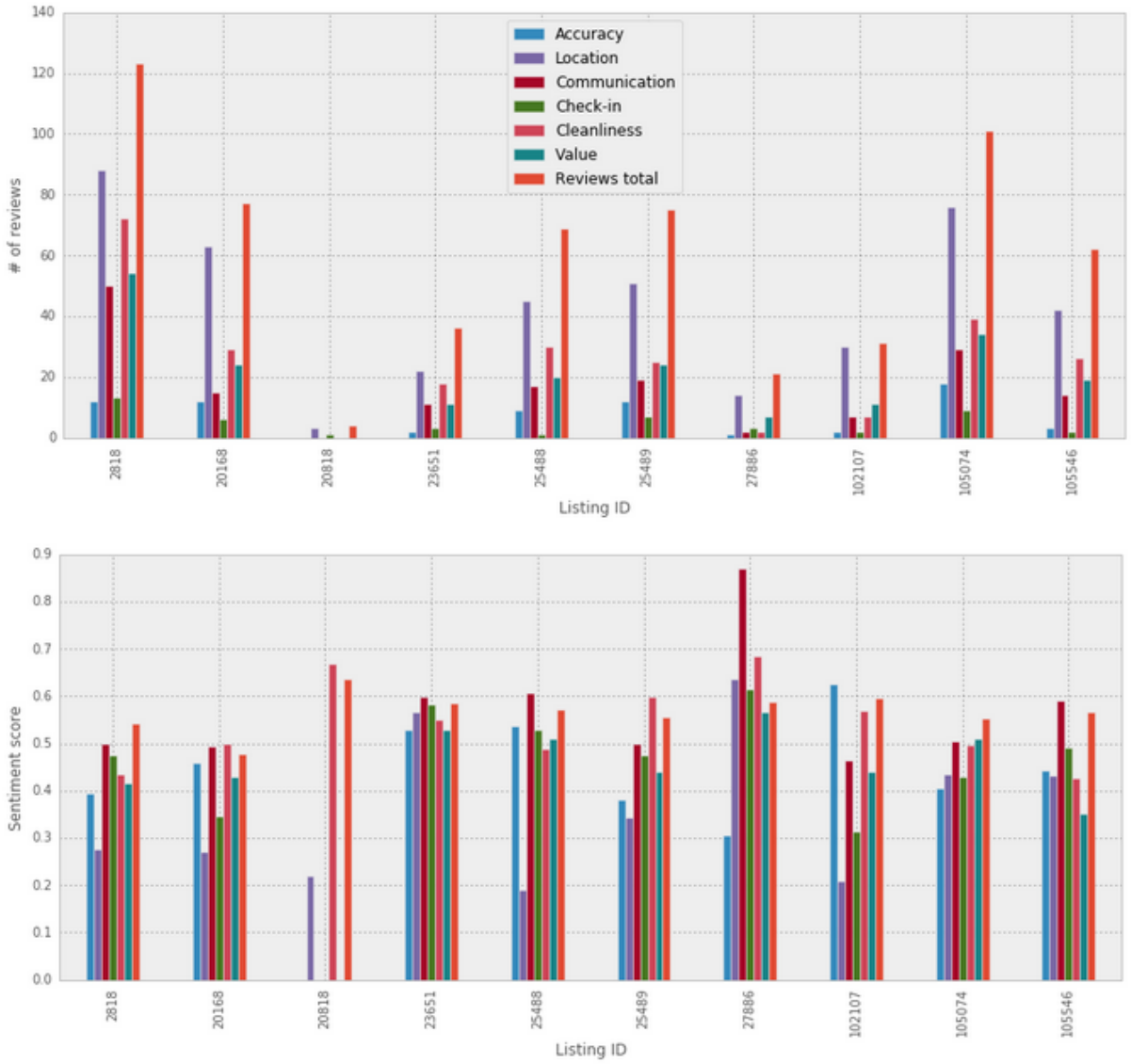


Figure 6.3: Features mentioned in review level per each listing and their sentiment

listing with ID 22886, even though it has the highest sentiment score on *communication*, only a very few people have commented on it, compared to the other listings. This result means that the sentiment score of the first listing would be more reliable than the one with a few reviewers but a good sentiment score. In this case each customer would have to make an individual choice for which listing they would go by considering both the number of reviews on a certain feature and their sentiment.

6.5 Pipeline results compared to Airbnb stars

An important aspect of analysis from the generated sentiment scores is their comparison with the Airbnb quantitative data. In the Airbnb feedback system, the customers can see the number of reviews for a listing, the overall average rating after the first three reviews and the average rating of the six specific features: *accuracy, check-in, cleanliness, communication, location, value*. These ratings are compared to the ratings generated by the pipeline by converted the sentiment scores into 1-5 stars as explained in the Methodology chapter. The comparison between pipeline generated stars and Airbnb is done for overall ratings and in feature level.

6.5.1 Overall stars per listing

From the analysis we can notice that the overall ratings generated from the pipeline are in general quite similar to the ratings in the Airbnb system. Both in Airbnb system and the pipeline results, there is no listing with negative sentiment, lower than 3 stars. All the possible combinations of values are retrieved from the dataset and they are presented in Figure 6.4. The most common combinations are (5:4),(5:4.5) and (4.5:4), where the first indicator belongs to Airbnb rating and the second to the pipeline. From the combination can be noticed the existence of a systematic tendency of Airbnb ratings to be higher than the ratings generated from the text feedback, concretely in 84.6% of the cases. These differences are presented in Table 6.2, from where we can see that 52.9% of cases have half a star difference, one star in 31.5% of cases and only 0.14% of the cases difference more than one star (3 cases explicitly).

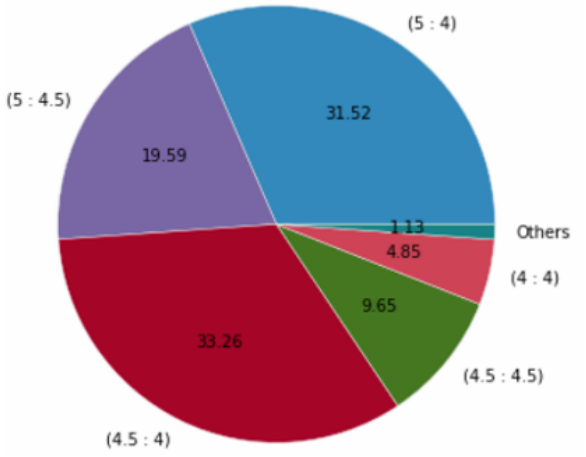


Figure 6.4: Comparison of combinations Airbnb and pipeline stars per listing

Difference	Frequency	%
0.5	1097	52.9
1.0	654	31.5
0.0	301	14.5
-0.5	17	0.8
1.5	3	0.1
-2.0	1	0.04

Table 6.2: Differences in stars between Airbnb and Pipeline

The differences more than 1 stars are treated as isolated cases and they consist of three listings with ID *1357971, 1410370, 2606699*, where Airbnb's ratings are 1.5 stars higher than the pipeline, and only one case with ID 1022631 of difference (-2). For this case, Figure 6.5, where the difference is the largest, we found out that the listing has actually only one review with one sentence, which has high sentiment score and where the pipeline is based for generating the overall

rating.

	Listing ID	Review ID	Sentence	Sentiment score
6515	1022631	10747505	Had a lovely break, very cosy home had most co...	0.771737

Figure 6.5: Reviews of listing with (-2) stars difference

For measuring the differences between values predicted by the pipeline and the actual Airbnb values, the chosen method is Root Mean Square Error (RMSE), which is explained earlier. For its estimation the Formula 5.1 is adapted to:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(y_{\text{i (Airbnb)}} - \hat{y}_{\text{i (Pipeline)}} \right)^2}{n}} \tag{6.1}$$

For the overall ratings, the RSME is 0.675 which means that a certain bias exists between the two samples.

6.5.2 Stars of features per listing

The same comparison as for overall ratings is repeated for each feature. In contrast to the results from overall ratings, the combinations of Airbnb and pipeline stars for features appear more distributed, new combinations are present and the differences between the predicted stars and Airbnb are bigger. Table 6.3 shows the differences in stars for two features, Accuracy and Cleanliness. Comparing the differences can be seen that for *Cleanliness* the differences are mostly 0.5 stars but for *Accuracy* difference of 1 stars prevails. From the observation it is noticed that differences can reach in quite a few cases higher than 1.5, which has two possible explanations. First, the pipeline is mistaken on generating a score per feature and second, the pipeline achieves to extract features with negative sentiment, which the system doesn't. The RMSE are calculated for each feature and it results that *Location* is the feature with the lowest RMSE, 0.48, and *Checkin* is the feature with the highest value, 1.51. From the analysis of feature mentioning we found out that *Location* was the most mentioned feature and *Check-in* the least mentioned one (Section 6.2). The comparison of RMSE values leads us to thinking that *the more the feature is mentioned or the more reviews a listing has, the more similar the values of the pipeline are to Airbnb*. Therefore, to test

Feature:Accuracy			**Feature:Cleanliness**		
Diff.	**Freq**	**%**	**Diff.**	**Freq**	**%**
1.0	649	41.2	**0.5**	839	47.1
0.5	571	36.9	**0.0**	486	27.3
0.0	162	1.4	**1.0**	286	16.1
1.5	105	6.7	**-0.5**	111	6.2
-0.5	32	2.1	**-1.0**	27	1.5
2.5	10	0.6	**1.5**	22	1.2
2.0	7	0.4	**2.0**	6	0.3
-1.0	6	0.3	**2.5**	2	0.1
3.0	5	0.3	**-2.5**	1	0.05
3.5	1	0.06	**-1.5**	1	0.05

Table 6.3: Differences for two features

this hypothesis the limit of reviews for a pipeline to generate a rating is set to 3, the same limit that the Airbnb uses for aggregating the average number of stars. This part of analysis is treated separately in the following section.

6.6 Conditional analysis on the number of reviews

The hypothesis that the number of reviews is reflected in the performance of the pipeline is tested in two levels: for the overall ratings and for each feature. In the first case, the condition set is that a listing must have at least 3 reviews for the pipeline to generate an overall rating. In this case the RMSE changes from 0.675 to 0.673, which is not a significant change. However, the cases with big differences are reduced from 4 to 2. On the other hand, in feature level looks like the impact is more significant. The condition in this case is set to "each listing must have the feature *mentioned* in at least three reviews for the pipeline to generate a sentiment score for it". Figure 6.6 shows the RMSE of differences in Airbnb and the pipeline before and after setting

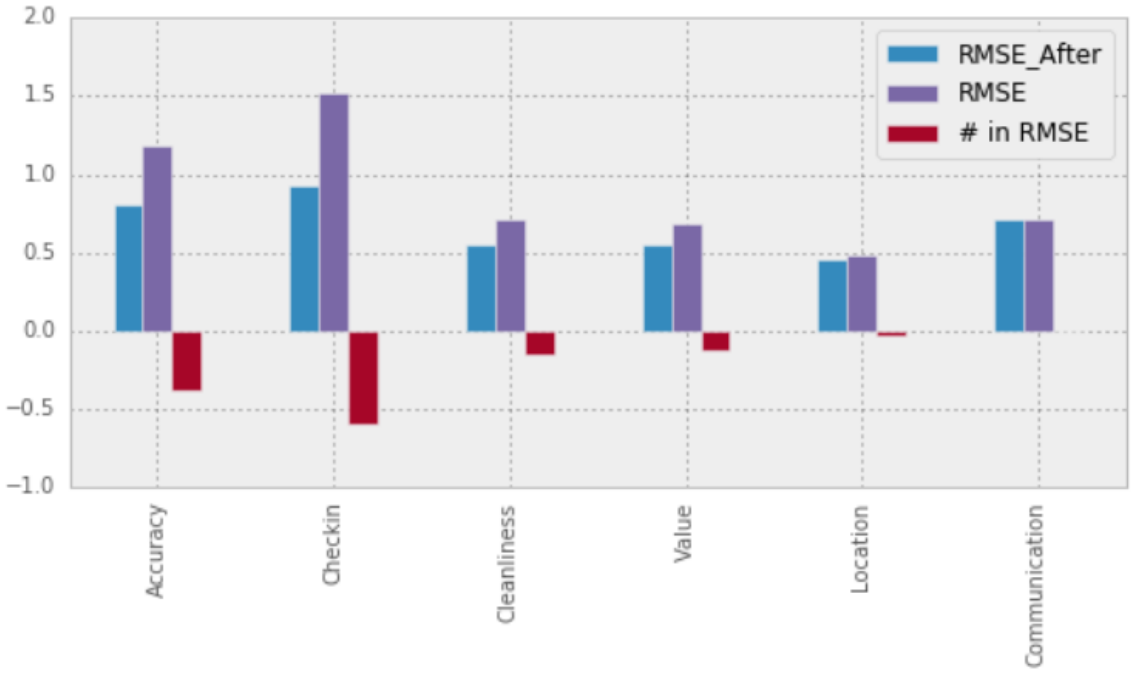


Figure 6.6: RMSE per feature before and after setting the condition

the condition of 3 reviews. The differences are quite significant for *Accuracy* and *Check-in*, which aligns with the features with the smallest number of reviews. Setting this condition makes the analysis of features more reliable and removes biggest the outliers. Figure 6.7 shows how the distribution of the combinations changes after conditioning for these two features. The details of this analysis and the cases of other features can be found in Appendix.

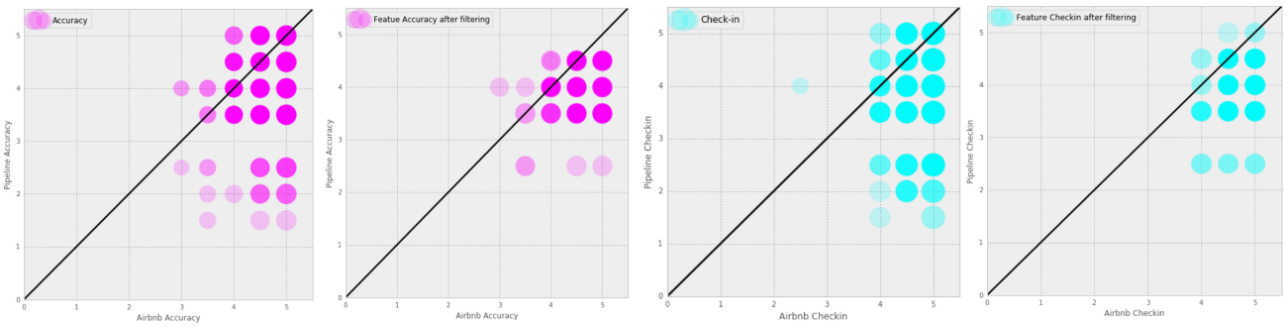


Figure 6.7: RMSE per feature before and after setting the condition

Secondly, from the analysis of star distributions per features as shown above it is noticed that the pipeline identifies features with negative sentiment better than the star rating system of Airbnb. In the first overview these cases are seen as outliers but even after conditioning the cases in 3 or more reviews per feature, the phenomenon persist. Some of the cases are investigated closely and they result on true cases where the reviewers negatively comment about features. On the other hand, this result is quite expected as the customers tend to write positive feedback about the listing in general, but explicitly point out things they don't like. This part of analysis is important for service providers as it identifies the features that they should improve.

6.7 Reducing the bias of Airbnb system

In their study on the existing bias in the Airbnb star system, [**?**] suggest text mining as a way for reducing the bias of the system. In a few words, the study indicated that the reviewers tend to highly rate the listings and only text reviews reveal the negative aspects of a listing [**? ? ?**]. In the previous sections we have mentioned the presence of higher rating from Airbnb system than the text reviews and we have proven that indeed text reviews reveal hidden knowledge about features that the system assigns high ratings. By calculating the mean in differences between the two samples, we suggest that by subtracting half a star from Airbnb values we can retrieve better estimators of stars for a listing, which will both reflect the actual ratings of Airbnb and the ones generated from the text reviews. The RMSE of differences between pipeline and the adjusted sample is 0.357, meaning that the differences are reduced significantly considering that initially RMSE was 0.675. Most importantly the RMSE is comparable with the standard deviation of the sample, which is 0.312. Thus, the values generated by the pipeline are an unbiased estimator of the overall star rating in the Airbnb system. In conclusion, the reduction of Airbnb values with half a star is recommended for retrieving ratings, which consider both manual ratings in the system and the text reviews.

CHAPTER 7

DISCUSSION

7.1 Major findings

Getting insights about features hidden in free text is an important step towards better customer feedback analysis. The proposed approach on the first stage extract features from text based on the accommodation ontology, secondly it detects the sentiment of each sentence and assigns it to the features mentioned in each sentence respectively. Thus, the pipeline returns all the sentiment scores of sentences and features, which are then grouped and calculated for reviews and listings. This work is most similar to [**? ? ?**] in the concepts discussed in the second chapter, however mining the text for quantitative analysis purposes is a novel approach.

As part of the main findings, this paper shows the alignment between the quantitative data of Airbnb and the output of the pipeline. Furthermore it suggests that there is a systematic tendency of Airbnb ratings to be higher than the ratings generated from text feedback. This tendency consist of half a star, which is recommended to be reduced from the Airbnb system ratings. In addition, we show that the reviewers are in general very positive in their text messages as indicated by their star ratings. However we showed that the text reviews can reveal hidden knowledge which can not be obtained only by analyzing the ratings in the feedback system. Thus, even though the overall ratings of a listing align between pipeline and Airbnb, in feature level it is important that we mine the text as it influences features' ratings. We found out that the most mentioned features in the reviews are *location, value* and *cleanliness*. Their sentiment is necessary that to be conditioned based on the number of reviews, where the feature is mentioned within a listing. The minimum number of reviews referring to a certain feature within a listing is set to 3, the same limit that Airbnb uses for calculating their average star ratings. With this

condition set, we found out that 16.5% of the listings do not have enough reviews for generating reliable scores about sentiment of features.

The analysis of user generated content in Airbnb system includes both customer and business perspective, therefore it is seen as an interesting aspect to explore further than this research. The evaluation of the pipeline showed that it successfully detects the sentiment of the sentences, and for the second task, that of feature extraction, the pipeline reached an average precision of 77.7% and recall 79.2%. However, for explaining possible downfalls of the proposed approach, the following subsection will describe the limitation and future work to be done for improving the pipeline and it will be followed by the practical implications of this work.

7.2 Practical implications

With the increase of user generated content, service providers and businesses deal with the challenge of reducing the costs of having employees to read through large amounts of text data such as surveys, Web documents, blogs, reviews,and social media sites in order to extract needed information. Having this analysis automated and detailed into discrete opinion scores reduces significantly these costs. Secondly, it gives to businesses a priority advantage of keeping track of a measurable impact that their product or services have in the crowds. In addition, it provides them a clear panorama on which aspects they need to improve, which aspects their competitors are doing better and also the aspects on which their services are superior in the market. Furthermore, having the insights of text reviews in the form of quantitative data allows businesses to customize the analysis and the insights they want to take out of this data. In the same time, these tasks can be performed by an employee who has no knowledge about opinion mining and they wouldn't need to, because the output is just a traditional format of quantitative data, suitable for analyze with Excel, SPSS, Matlab and so further.

From the customer point of view, having the service owners better understood their needs, they will have better services. Furthermore, users will get encouraged to write more reviews as their voice with be heard, opposite of the case when a review ends up in the 5th page of a listing and it is read by less than 3% of other users [**?**]. In addition, if the ratings generated from text will complement the star rating system as suggested, the results would be more reliable and users' work on would be reduced. Finally, with the right implementation of the pipeline in search engines the customers will be able to directly rank the results based on their required features.

7.3 Limitations and future research

This study has several limitation, which create room for further research. Due to the time-frame this research is based only in six accommodation features that are mentioned in the Airbnb website. In order for the study to be complete every feature of the accommodation ontology will have to be treated. This step would give meaning to all the sentences of reviews by clustering

them into one or more features and would decrease the phenomenon of having too many sentences with un-identified objects. Secondly, this research groups all the accommodation features in the same family root without making a distinction between explicit and implicit features. The last ones are more difficult to process in NLP, however different academics have taken steps in this direction. A N-gram approach is very important to be tested and it is believed to improve the performance of the pipeline for feature identification. For example in phrases like *"The room was **as in** the **pictures"*** is a potential 3-gram phrase which refers to *accuracy*. Thirdly, for sentiment detection the paper discusses the use of VADER, however several algorithms (i.e. SO-CAL, SenticNet, Pattern) can be trained in the corpus for performance comparison purposes.

In addition, the evaluation of the pipeline requires higher values for TP/FP which affects the reliability of the precision and recall measurement. Thus, the evaluation will have to be improved either by increasing the number of features identified as discussed above or by increasing the number of sentences in the evaluation set, so we will have more cases to evaluate. Lastly, the pipeline ignores all the reviews in languages other than English. In the Amsterdam's dataset, around 17% of the whole reviews ignored, meaning that from 30.2 reviews in average per listing, 5.1 reviews in average will be ignored. The ignored reviews are written mostly in Dutch, but also other languages. This percentage is expected to increase in cities other than Amsterdam. Therefore, the pipeline has to be trained for languages other than English, such as Dutch, French, Spanish and German, which correspond with the countries of biggest Airbnb impact in Europe. For doing so, the accommodation ontology has to be adjusted to these languages and secondly the sentiment detector must support those languages. However, the limitation on language and sentiment detection are part of the limitations of every tool which deals NLP, as new and non consolidated field of research [**?**].

Appendix A

Begins an appendix