Finding a Needle in a Haystack of Reviews: Cold Start Context-Based Hotel Recommender System

Asher Levi[*] , Osnat (Ossi) Mokryn[†]
School of Computer Science
Tel Aviv Yaffo College
Israel
asherlv2@gmail.com, ossi@mta.ac.il

Christophe Diot, Nina Taft
Technicolor ltd.
Paris, France, Palo Alto, USA
christophe.diot@technicolor.com,
nina.taft@technicolor.com

ABSTRACT

Online hotel searching is a daunting task due to the wealth of online information. Reviews written by other travelers replace the word-of-mouth, yet turn the search into a time consuming task. Users do not rate enough hotels to enable a collaborative filtering based recommendation. Thus, a cold start recommender system is needed.

In this work we design a cold start hotel recommender system, which uses the text of the reviews as its main data. We define context groups based on reviews extracted from TripAdvisor.com and Venere.com. We introduce a novel weighted algorithm for text mining. Our algorithm imitates a user that favors reviews written with the same trip intent and from people of similar background (nationality) and with similar preferences for hotel aspects, which are our defined context groups. Our approach combines numerous elements, including unsupervised clustering to build a vocabulary for hotel aspects, semantic analysis to understand sentiment towards hotel features, and the profiling of intent and nationality groups.

We implemented our system which was used by the public to conduct 150 trip planning experiments. We compare our solution to the top suggestions of the mentioned web services and show that users were, on average, 20% more satisfied with our hotel recommendations. We outperform these web services even more in cities where hotel prices are high.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval - *Information filtering*

General Terms: Algorithms.

Keywords: Recommender systems, opinion/text mining, context-aware recommender systems, common traits, sentiment analysis.

1. INTRODUCTION

The Internet has overtaken word of mouth as the primary medium for choosing destinations [1]; 63% of consumers plan travel by searching the Internet, visiting an average of 22 sites before deciding on a destination.

Producing recommendations for travel is inherently difficult, as an individual rarely rates more than a small number of hotels and thus rich profiles cannot be built. Having limited or no information about the user translates to a user *cold start* recommendation [2, 3, 4]. An intelligent cold start recommender will minimize a new user's effort while still learning enough to recommend the user a product that is likely to be of her interest.

In this paper we design a context-based search recommender system. We show that context-based search can be facilitated for the construction of cold start recommender systems. We further show that contextual information can be mined from review texts, and analyzed for *common traits*[1] per context group. Contextual information has been newly recognized as an important feature when the consumer decides to make a purchase [5, 6, 7]. A lot of research has already been performed in the area of recommender systems and information retrieval. However, most recommender systems focus on recommending the most relevant items to users without taking into account any additional contextual information. Most existing information retrieval systems base their retrieval decisions solely on queries collections, whereas information about search context is often ignored [8].

Users' search patterns are context-based.Among the plethora of reviews, readers opt for recommendations from travelers with comparable needs. A single traveler may share the same needs as other single travelers. A user traveling with her family has different needs from a user traveling on a business trip, i.e. the user context information is an important factor in choosing a hotel. When a user reads reviews she can metaphorically be seen as wearing personalized glasses. Reviews are read through those glasses, and particular words or comments will resonate, positively or negatively, with the reader based upon her needs for her upcoming trip and her personal preferences. Special attention is often given to reviews written with the same intent, or by reviewers from a comparable background. Hence, we define three types of context information. The first is *intent*, or purpose of the trip. We include 5 categories of intent, namely *business trip, single traveler on vacation, family, group, couple*. The second is *nationality*. The third context is user preferences for the different hotel aspects. These were mined from the text using an unsupervised clustering algorithm. We tagged the different clusters found in the text as *location, service, food, room, price-value quality* and the *facilities* (pool, spa, etc). Thus, a user using our system is asked to provide her trip intent, nationality, and preferences for these aspects.

[*]Part of this work was done while Asher Levi was visiting Technicolor lab in Palo Alto.
[†]Corresponding author

RecSys'12, September 9–13, 2012, Dublin, Ireland.

[1]In psychology, individuals are often characterized by cardinal traits, while groups (such as nations) can be characterized by common traits.

We obtained data from Venere.com and TripAdvisor.com. The database contains details for each hotel: the hotel's general information, reviews and ratings. In a pre-processing phase, we mined the text and found *common traits* for each context group. These are found in the form of typical words that appear more in text written within that context but are not common for other contexts. A clustering was used to group words that refer to each aspect. Thus, at the end of this pre-processing phase, we have significant words per context, be it intent, nationality, or hotel aspect.

Our recommender system mines the text of the reviews similarly to the user wearing personalized glasses. The user is prompted for her trip intent, nationality and preferences per hotel aspect. We introduce a novel weighted algorithm for context-based text mining. The core idea of the algorithm is to give more importance to reviews of people with the same contexts as the user's. Common traits per the user's context groups and words that describe favorable hotel aspects are given a higher score than other words. We further find the sentiment expressed in the review per context (i.e., positive or negative) and give a corresponding score. Thus, the final score for each review corresponds to that of a user's with comparable needs and preferences and coming from a similar background.

We implemented our system and published it for use, presenting to the users results from our system combined with the top suggestions of the above mentioned web sites. We had over 150 evaluations by friends and colleagues who looked for hotels in four major European cities. Hotels recommended by our system were favored (60.2%) compared to TripAdvisor's and Venere's top suggestions (50.8%). More significant is the fact that our raters said they would not stay in 26.4% of the top hotels recommended by these sites, whereas with our context-based recommender, their dissatisfaction was much lower at 15.9%.

The contributions of our paper are the following: We designed a hotel recommender system that outperforms current leading web sites top suggestions; We define context-based search as a method to overcome the cold start problem for users; Our system is the first we know of that relies mainly on the text of reviews for a cold start recommendation. Hotel ratings and groups' bias are used as tie breakers; We devise a weighted text mining algorithm that leverages common traits found per context group to enable the processing and evaluation of text per users' needs; To find hotel aspects we use a community detection algorithm that leverages the spin glass theory, and changed its distance function to account for the extra clustering overlapping exhibited in the text. This enabled us to find the different hotel aspects in an unsupervised fashion; We held experiments with Mechanical Turk workers and showed that reviews are perceived differently than ratings given by the reviewers, suggesting that sentiment analysis of the text cannot rely on the ratings, although commonly used.

2. RELATED WORK

One of the common and difficult problems for recommender system is the *cold-start* problem, a situation in which the system needs to recommend a product to a new user that has no past information or a new item with very few or no rating [2, 4, 3]. We build a model for domains that by nature don't have a lot of history (or not at all) information about the user, and user cold start recommender system is required.

Despite the abundance of studies targeted at solving the new item problem [4, 9], there has been little work in solving the new user problem. The dominant approach is using a learning phase, in which a user is asked to provide a set of ratings for selected items, in a way that gathers as much information about the user as possible [10]. Another approach presented in [11] exploits the significance of users' implicit feedback for alleviating the new user problem. In this approach the user has to express interest in items, or organize the items in relative order, without providing explicit ratings for those items. Those approaches use only ratings or relative rankings on items and thus are bounded under a rating recommender system limitations. Moreover, they are missing all the information that can be extracted from the text.

Another approach is the "Metadata" approach; here the metadata of an item is used to create content-based recommender systems. This method relies on systems where the user needs to provide some demographic data. The solution presented in [12] is utilizing the strength of the vector aspect model with user information; they used the demographic information of the user (age, gender and job) as the user's features. A model of relationships between a user's demographic information and an item's metadata was presented by Park et al. [13]. Those solutions use only ratings at their model; They use the user's context information as a feature for building their recommendation; We extend the usage of contextual information. We are not considering only the user general information (e.g, age or gender) and simply profile the user, but rather we are trying to build a more complete behavioral profile that attempts to capture the expectations of the user from our recommendation;

By using a context information that relevant to the current session of search (e.g, for hotel recommendation we use the intent of the trip as one of the context features), we are capturing a more accurate and efficient profile.

3. SYSTEM OVERVIEW

A hotel recommender system typically won't have sufficient historical information to build profiles for individuals. It does, however, have additional data in the form of reviews that is sufficient to enable the characterization of context groups. We give here an overview of our system, which determines *common traits* for groups that share the same context. The core idea of our system is to give more importance to reviews of people with the same context. Our system brings greater importance to the topics those reviewers focus on frequently and also focuses on topics that are associated with the user's stated preferences.

In the hotel arena people can be categorized by their trip intent (such as those who travel as a 'couple', or a 'family', etc.) and nationality, which we refer to as context groups. Using the text reviews from multiple people within a single context group, we can essentially find the common traits of groups such as 'family' travelers (and so on for the other categories). We additionally process the corpus of reviews to identify the vocabulary that is used to describe a particular aspect of a hotel. Once a person using our system specifies her intent, nationality and preferences our system evaluates reviews with accordance with the traits and preferences, and gives a recommendation.

We now give a brief overview of the components and steps of our method, depicted in Figure 1. The top 3 boxes on the left correspond to the pre-processing phase in which we define the common traits of intent and nationality groups, and define the different hotel aspects referenced in reviews, correspondingly. To find common traits for each context group, we extract the nouns and noun phrases (called *features*) from all reviews and find those that are more common for that group. These features are then assigned a weight per each context according to their relative frequency in reviews within that context. The higher the weight, the more important a feature. The common traits of context groups are the higher weight features for that group. Hence, the common traits of Italians consist of a set of features and their weights, while these of Germans may contain largely the same features but with different weights. Common traits

of hotel aspects are constructed differently. Here we carried out a clustering task to cluster features based upon co-occurrence in the same sentence. Each feature can only occur in one cluster, and thus each cluster contains the most relevant vocabulary for that aspect. The fourth component of the preprocessing consists of building an opinion lexicon which will allow us to analyze adjectives associated with features, and to give each feature an orientation score depending upon how positive or negative is the sentiment of any associated adjectives.

While the base weight of each feature is one in our system, features that are distinctive of several context groups may have different weight per group. The specific set of weights used in response to a user hotel search will be chosen once the user declares her context and preferences. For example, if a user specifies 'business traveler' as her intent and her nationality, then the set of feature weights used will be those in the 'business traveler' group and the corresponding national group. In our figure, this step corresponds to the "select relevant feature weight for intent" and ".. for nationality" boxes. Similarly, corresponding weights are given to features of important aspects. This implies, for example, that the feature 'air conditioning' will get one weight depending upon its importance for business travelers, a second weight depending upon its importance for the given nationality, and a third weight depending upon its importance per the user preference for the aspect it belongs to. The final weight for each feature is done by combining these three weights (depicted as "build feature score" in the figure).

Next we use our opinion lexicon to give each feature an orientation score. We subsequently combine the features, their weights and orientations to build a score for each sentence. The sentence scores are then combined to give an overall score for each review. This score should reflect the relative importance of the given review for the user. Reviews that are both important and positive are deemed most relevant thereby receiving the highest scores. The final score for each hotel is an average of all of its reviews, each of which is scored from the user's perspective (i.e., based on her context and preferences), and an adjustment bias calculated per the context given.

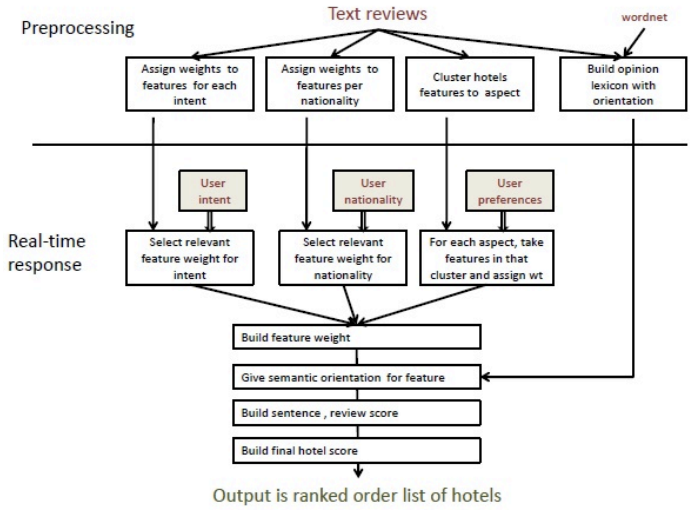

Figure 1: Overview of our approach

4. CONTEXT BASED ANALYSIS

The main idea of our algorithm for context based text analysis is to assign weights to common traits per context. Thus, at the end of the process, each review is mapped to a score number, based on system's perception of the user's perspective.

4.1 Intent and Nationality Profiling

We find common traits for each context group by mining the text from reviews on a sentence level. Our approach is to extract key features (i.e., words) that are important for each group. It has been shown that a reviewer's vocabulary when commenting on an item was found to converge, in the sense that the most frequently used nouns and noun phrases used correspond to genuine and important features [14]. Similar to [15] we extract features and remove redundant and meaningless items from the candidate features we found.

The basic building block of our algorithm is the trait based weight assigning. For each review written we extract the features and assign each feature a weight that reflects its importance for each context group. Let c denote a general context that can be either an intent (or purpose) p or a nationality n (i.e., $c \in \{p\} \bigcup \{n\}$), and let $freq_f(c)$ denote the frequency of feature f for context c. The frequency of a feature per context is the relative number of occurrences of feature f in sentences appearing in reviews that belong to context c. For example, the frequency of the feature 'WiFi' for Americans is calculated as the ratio of the number of times this feature appeared in sentences written by Americans, divided by the total number of sentences written by Americans. Similarly, avg_f is the average frequency of feature f, $stdv_f$ is its standard deviation, and $dev_f = avg_f - stdv_f$. Using this notation, we define the weight of a feature f for a given context as follows:

$$W_c^f = \begin{cases} 1, & \text{if } |dev_f| < stdv_f \\ Max(0.1, 1 - \frac{dev_f}{stdv_f}), & \text{if } \frac{dev_f}{stdv_f} < -1 \\ 1 + \frac{dev_f}{stdv_f}, & else \end{cases} \quad (1)$$

The majority of features will either be assigned a 1; however those whose frequency is larger than average plus or minus one standard deviation, are assigned values between 1 and 3 or 0.1 and 1 respectively. Hence each feature is assigned a weight in the range $[0.1, 3]$ per context.

4.2 Aspect Profiling

Recall that we ask the user to input their preferences on six aspects. These aspects were not selected at random, but were instead the result of a word clustering analysis we performed on the text. Often in reviews, different words may be used to refer to the same general aspect of a hotel. For example, words like 'area', 'street', and 'metro' may all refer to aspects of a hotel's location. There are many approaches to clustering, hierarchical clustering, partition clustering (e.g k-means) etc. The number of clusters, k, is usually either an input parameter or found by the clustering procedure itself [16]. In our case, clustering would yield the different hotel aspects and therefore should not be supervised but determined by the clustering algorithm over the text itself.

To account for the sparsity and the overlapping characteristics in the network of word features, we build upon an unsupervised community detection[2] [17, 16] technique based on [18]. We build a network graph in which each node corresponds to a feature and each community will correspond to an hotel aspect. Trying to find the maximal modularity is defined as finding a partition that will minimize the energy of the features network graph. The Hamiltonian, denoted in equation 2 is defined in the following way: exist-

[2]We use the terms cluster and community interchangeably

ing internal edges and non-existing external links (between formed communities) minimize the Hamiltonian, while existing external links and non-existing internal links increase its value. The algorithm tries to find a partition that minimizes the Hamiltonian, based on the spin glass model for finding a partition that minimizes the energy of the spin glass with the spin states being the community indices.

$$\mathcal{H}(\{\sigma\}) = -\sum_{i\neq j} a_{ij} \underbrace{A_{ij}\delta(\sigma_i,\sigma_j)}_{\text{internal links}} + \sum_{i\neq j} b_{ij} \underbrace{(1-A_{ij})\delta(\sigma_i,\sigma_j)}_{\text{internal non-links}} \quad (2)$$
$$+\sum_{i\neq j} a_{ij} \underbrace{A_{ij}(1-\delta(\sigma_i,\sigma_j)}_{\text{external links}} - \sum_{i\neq j} b_{ij} \underbrace{(1-A_{ij})\delta(\sigma_i,\sigma_j)}_{\text{external non-links}}$$

Where A_{ij} is a boolean adjacency matrix, $\sigma_i \in 1, 2, \ldots q$ denotes the indices of the communities, with q the number of maximal communities. [19] showed that the division does not depend on q, for large initial q values.

In [19] a_{ij} and b_{ij} where chosen as a function of the probability of two graph nodes to be adjacent under the assumption that when this probability is high the nodes are more likely to belong to same group, or community. In our case, this translates to the probability of two features to appear in a sentence together. However, we found in reviews that very frequent features are often found in sentences together. For example, it is common to find sentences of the following structure:

The location was great and the room was very clean.

Clearly, location and room belong to different hotel aspects, and should therefore belong to different communities. To account for this tendency we instead use the PMI-Pointwise mutual information weight, which measures the information overlapping between two random variables [20], described in (3).

$$PMI_{ij} = \log\left(\frac{p(i \wedge j)}{p(i)\cdot p(j)}\right) \quad (3)$$

Where, $p(i)$ is the probability that the feature i appears in a sentence. Then, $a_{ij} = \gamma \cdot PMI_{ij}$, where γ is a parameter expressing the relative contribution to the energy from existing and missing edges. In our case we chose $\gamma = 1$.

Over the corpus of reviews, our PMI-pointwise improvement of the spinning glass community detection algorithm produced six clusters of different sizes (note that the number of clusters is unsupervised). The identification of these 6 clusters is important as it determined the particular hotel aspects that we chose to ask users their preferences for. Each cluster and the set of features it contains can be thought of, intuitively, as the *common traits* for the aspect associated with this cluster. These clusters are useful as follows. Suppose for example that a user specifies that location is of utmost importance to her. The room cluster identifies a large number of features (or words) that are often used to discuss things inside a hotel room; thus reviews in which these words occur frequently are more important to a user who cares about the room than one who cares about food. After studying the words that ended up in each cluster, we selected the cluster names as indicated in Table 4.2. These clusters can be computed ahead of time as part of the system's preprocessing.

Our weight assigning algorithm for aspect related features relates to the user's preference and is calculated online as follows: Let $u_{pref}(k)$ denote user u's preference for aspect (i.e. cluster) k. If feature f is in cluster k, then we calculate the weight for the feature according to the users preferences as follows:

$$W^f_{u_{pref}(k)} = 1 + \frac{u_{pref}(k)}{5} \quad (4)$$

where, $W^f_{u_{pref}(k)}$ denotes the weight of feature f for user u according to her preference $u_{pref}(k)$. For example, if the user sets their preference for location to 5, and the feature is *train*, then the weight of train for this user is 2. Another user that specifies that location is of importance 1, would have the feature *train* assigned a weight of 1.2. When determining the weight for the feature *train* we only use the user's preference for location (and not for room or food) because the feature 'train' is in the location cluster and cannot be in any other cluster.

4.3 Feature Opinion Orientation

Next we determine the polarity of the opinion expressed in the review on each feature, whether positive or negative, to assign a corresponding sign to a feature's weight. To infer the opinion polarity per feature we use an opinion lexicon. An opinion lexicon is a dictionary of words and word phrases that express positive or negative sentiments. In this work, we consider sentiment words to be adjectives the reviewers use to express opinions on product features, as in [21, 22] . To collect the opinion word list we use a corpus-based approach similar to the approach described in [14, 23]. We extract all the adjectives that appear in the same sentence for each feature.

We then find the semantic orientation of the extracted opinion words. When the reviewer uses a word that expresses a desirable state, then the word is classified as having a positive semantic orientation. Similarly, an undesirable state translates to a negative semantic orientation. We use a bootstrapping lexicon-based approach as in [14]. Manually, we create a set of seed adjectives from the opinion lexicon list with semantic orientation. Then for each adjective in the seed list, we search for a synonym and an antonym in WordNet [24]. Each found adjective in the opinion lexicon is assigned an orientation, and is added to the seed list. The seed list grows in the process. A recent work [25] suggests to consider the influence of aspects on sentiment polarity. However, given that we give the weight per feature and that the aspect counts only for a fraction of the total weight we left the orientation per feature as before.

We used common opinion rules as described in [23]. One is the negation rule, words or phrases like 'no', 'not', etc. take the opposite orientation expressed by the opinion phrase. The other is the *But* clause rules, a sentence containing 'but' also needs special treatment. The opinions before and after a 'but' are usually opposite of each other. First we try to determine the semantic orientation of the feature in 'but' clause. If we cannot get the orientation of the phrase we take the opposite orientation of the clause before the 'but' clause. Phrases such as 'with the exception of', 'except for' etc. behave similarly to 'but' and are handled in the same way. For example, in the sentence "The room was clean except for the bathroom", the opinion about the feature *room* is positive and the feature *bathroom* gets the inverse opinion which is negative. There are also some phrases that contain negation and but words, yet do not change the orientation of the opinion. For example in the phrase "I do not only like the size of the room, but also its style", the 'not', 'but' words do not change the orientation of the opinion words 'like' and 'style'.

Using these rules and our lexicon, we assign an orientation score to each feature f in a given sentence s, denoted $score(f, s)$. It should be clear that the same feature, in two different sentences, could receive different orientations. When many opinion words surround a single feature, they are aggregated as indicated in equation (5).

$$score(f, s) = \sum_{op\in s} \frac{or_{op}}{d(op, f)} \quad (5)$$

Aspect tag	Features per aspect (randomly chosen)
Location	location, area, city, street, ,metro, station, train, distance, bus, airport
Service	staff, service, hotel staff, reception, front desk, luggage lobby, reception, staff, person, wifi
Food	breakfast, morning, food, restaurant, bar, coffee, buffet ,dinner, fruit, terrace, buffet variety, bread, course
Room	bathroom, floor, shower, size, window, door, view, building, tv, water, elevator, balcony, hotel room, lift, bath
General	hotel, night, place, stay, price, experience, trip, value, hotel star, money, rate, money value, deal, quality, cost
Other	pool, spa, gym

Table 1: Spin Glass community detection algorithm results

Here op is an opinion word in sentence s, $d(op, f)$ is the distance (word count) between feature f and opinion word op in sentence s. Also, or_{op} is the orientation $(-1, +1)$ of the opinion word op. Dividing by the distance between the feature and the opinion word is used to give lower weights to opinion words that are farther away from f. When the final score is positive, then the overall opinion of feature f in s is positive, and similarly the reviewer's opinion of the feature is negative when the final feature score is negative.

4.4 Producing a Review Score

We now have a set of weights and their orientation per the user's context for each feature in a review. We combine these elements to produce a single score for a review as follows. Given the user's input on their context, each feature has 3 weights, one for intent, $W^f_{u_p}$, one for nationality $W^f_{u_n}$, and one based on aspect preferences $W^f_{u_{pref}}$. The final weight W^f_u assigned to feature f for user u is the multiplication of these three weights, namely:

$$W^f_u = W^f_{u_p} \cdot W^f_{u_n} \cdot W^f_{u_{pref}} \tag{6}$$

The weights for each context are multiplied because that allows fine grained differentiation of people within our various groups (such as intent and nationality). Consider a Japanese person who uses our system. Based upon our nationality profiling, we see that the feature 'bath' is important. If that person also marks 'Room' as a hotel aspect that is very important to them (i.e. a preference of 5), then the quality of the bathroom is more important for this user than for a second Japanese person who marks 'room' as low priority and 'food' as high priority. This allows us to differentiate within nationalities by using the intent and preferences (or to differentiate within an intent group by their nationality and preferences).

To produce a score for each sentence, we multiply each feature by its orientation score and sum up the weight scores of all features in a sentence s, namely $\sum_{f \in s} W^f_u \cdot score(f, s)$. Similarly, we sum up the scores of all the sentences in a review to produce a score for a review v, as follows:

$$score(v, u) = \sum_{s \in v} \sum_{f \in s} W^f_u \cdot score(f, s) \tag{7}$$

Where $score(v, u)$ is the score of review v for user u. The review score captures how important a particular review is for the user based upon their context and preferences.

4.5 Devising a Hotel Score

Next we produce a score for each hotel so that hotels can be ranked and presented to the user in order from highest score to lowest. The major factor in the score of a hotel in our system is the score calculated for reviews based on user context groups and preferences. We term this the hotel orientation score, ho_u, where $ho_u = avg_{v \in R_{(h)}}[score(v, u)]$ and $R_{(h)}$ denotes the set of reviews for hotel h. The second argument is a bias adjustment, denoted b_{hsn}, which captures the bias of a user with intent p and nationality n, as well as any hotel bias h. (The bias term is explained below.) Thus our final hotel score is given by:

$$S^h_u = ho_u + b_{hpn} \tag{8}$$

Bias Adjustment to Hotel Score: In our hotel score, the orientation score coming from the text analysis of the reviews is the dominant component of the score, as these values will range from -40 to 80 approximately. Our bias terms range from 0 to 5 and are included primarily to break ties, or to differentiate hotels when their scores are very close. The process of using the star ratings needs to be adjusted for bias because there are systematic tendencies for some traveler groups to rate higher than others. For example, our data analysis shows that reviewers from Spain tend to rate lower in star rating systems than reviewers from the USA.

We compute the bias b_{hpn} for hotel h from traveler with both intent p and nationality n, as follows. Let μ denote the overall average star rating of *all* hotels in the system. The parameter b_h specifies the observed deviations of hotel h from the overall average. We use b_{hp} to denote the observed deviations that travelers with intent p have for hotel h, (and similarly for b_{hn}). These deviations are with respect to the average score of hotel h.

$$b_{hpn} = \mu + b_h + b_{hp} + b_{hn} \tag{9}$$

The average deviations are shrunk towards zero by using the normalization parameters, $\lambda_1, \lambda_2, \lambda_3$, which are determined by validation on the test set. For each hotel h we set:

$$b_h = \frac{\sum_{r_h \in R_{(h)}} (r_h - \mu)}{\lambda_1 + |R_{(h)}|} \tag{10}$$

where $R_{(h)}$ is a set of reviews for hotel h, and $\lambda_1 = 30$. The bias of intent group p for hotel h is:

$$b_{hp} = \frac{\sum_{r_{hp} \in R_{(hp)}} (r_{hp} - \mu - b_h)}{\lambda_2 + |R_{(hp)}|} \tag{11}$$

where $R_{(hp)}$ is a set of reviews for p and h, and where $\lambda_2 = 5$. The bias of a nationality n for hotel h is given by:

$$b_{hn} = \frac{\sum_{r_{hn} \in R_{(hn)}} (r_{hn} - \mu - b_h)}{\lambda_3 + |R_{(hn)}|} \tag{12}$$

where, $R_{(hn)}$ is the set of reviews from nationality n for h, and $\lambda_3 = 5$.

5. VALIDATION

The dataset used in this study was extracted from two well-known travel search engines, namely, Tripadvisor.com and Venere.com. For each hotel the data contains general information about the hotel (e.g, name, address, average rating, stars, price etc.) and a list of reviews written by hotel guests. The reviews include: travel intent of

the reviewer, nationality, rating, review text, and additional metadata. The data was collected for 4 cities in Europe: Munich and Berlin in Germany, and Milan and Rome in Italy. The data includes reviews that were written before January 2011. $84,968$ reviews were collected for $1,930$ hotels from TripAdvisor, and $52,266$ reviews for $1,845$ hotels from Venere. For each intent group we obtained thousands of reviews, from $6,541$ reviews written by people on business trips to $60,113$ reviews written by couples. Overall we collected information for $3,775$ hotels corresponding to $137,234$ reviews.

Recent researchers have turned to text analysis as there is potentially a great deal more information that could be extracted from reviews then star ratings[26, 27, 28]. In these solutions, the star rating is taken as the overall product rating, and additional information is obtained by analyzing the review text to extract opinions on specific aspects of each item, and thus improve a personalized recommendation. In [29] the authors incorporate a notion of personal scale, that is based on the observation that different users give different values to their describing words to improve sentiment analysis for personal recommendations. Our work differs from these because they use text reviews to build individual profiles and personal scales, thereby not focusing on the cold start problem. In the absence of user profiles, we choose instead to use the text to find the common traits of intent groups and nationalities, and further differentiate hotels by personal preferences.

The underlying assumption of [29] and many others, is that a user's ratings and text correlate. This assumption has not been previously quantified. We further wanted to quantify whether the perception of the reviews differs across context groups.

To this end we designed the following online experiment with Mechanical Turk workers. Each worker was given the text of five different hotel reviews. We used reviews from 50 different hotels' taken from venere.com, where ratings are on a scale of $[1-10]$. We then asked the workers, based on the text, to estimate the rate that the reviewer gave. To get meaningful results, we filtered out manually cases in which it was clear the workers did not read the text before estimating the rating[3]. We obtained $50-75$ estimates of the star rating for each review, yielding a total of 3715 estimates.

First, we checked whether the workers estimated the ratings were indeed similar to those given by the person writing the text. For each estimate, we computed the difference between the estimate rating and the actual rating, depicted in Table 5 . We averaged all these differences in cases when the estimates where higher than the actual, and computed a second average across all cases when the estimates were lower than the actual ratings. There is a clear skew between the estimated rate and the real rate. The skew is more significant when reviews were perceived as negative by the workers (1.67 more negative vs. 0.94 more positive). Two conclusions are thus possible. The first claims that users give an accurate star ratings while signifying negative aspects when writing down reviews. The other suggests that users are generous with the star ratings while expressing their real opinion in writing. In either case, these result indicate that star ratings do not consistently capture the sentiment in text reviews, and thus the correlation between text and reviews is weak.

To validate the assumption that context matters, we return to our Mechanical Turk experiment and ask whether the perception of the reviews differs across intent groups. For example, we saw that single travelers tend to rank almost the same as others but their text was perceived as much worse by the Mechanical Turk workers. The average difference between the perceived rate and the real rate was 1.93 on the average for single travelers. For people traveling in groups, the average difference between perceived and real rate was 1.54. This means that the text reviews of single travelers gives the perception to others of being far more negative than their corresponding rates; whereas for group travelers, the text reviews are only slightly more negative that the ratings would indicate[4].

Type	Average Difference	Reviews
Estimation > Rate	0.94	1474
Estimation < Rate	1.67	2241
Total	1.38	3715

Table 2: Mechanical Turk results, estimating the review's rate.

5.1 Data Analysis

To validate our algorithmic approach that takes into account context we asked ourselves whether different context groups rank hotels differently, whether the intent groups emphasize different things, and whether the tone of the reviews differs across context groups. In our case the different context groups are the different intent groups and different nationalities. The analysis was performed on reviews taken from TripAdvisor, where the hotel ratings are in the range of $[1..5]$. Table 3 shows the average rating for each context group, whether an intent group or nationality. Indeed we see differences across the groups, with the difference between intent groups varying to up to 0.55 (on a five star rating system). Next we computed frequent words used by each of the groups over both datasets, TripAdvisor and Venere. We removed words that are used by all groups. This left us with examples of words that were frequent to one group but not others. Table 3 also shows examples of words that are frequently used by one group, but infrequently or never, appear in the text of other groups. This indicates that the intent of a trip influences the content of reviews that get written, and that the top words (i.e. topics) that interest reviewers also differ by country or culture.

5.2 Tiebreakers Evaluation

The effect of considering the intent of a trip is clear both intuitively and from our results. We wanted to further verify the usefulness of using the nationality, as well as the bias factor used for the final hotel score. The following evaluations were done on our implemented system.

[3]We declined payments when the results seemed random, i.e., the worker did not seem to read the reviews at all. We had one dispute that was ruled in our favor.

[4]In a complementary experiment the workers were asked to estimate reviews' ratings when knowing the intent of the trip. The additional information did not affect the estimations, indicating the above results could not be predicted.

Context	Rate	Typical words
Family	4.15	Air condition, Car, Space, Shuttle, Breakfast
Couple	4.08	Coffee, View, Balcony, Breakfast
Group	4.02	Bar, Money, Bus stop, Shopping, Party
Single	3.8	WiFi, TV, Price, Supermarket
Business	3.6	Internet, Buffet, Park, Bar, Shopping, TV
U.S.A	4.11	Hotel staff, Train station, Lobby, Shuttle
Russia	4.07	Furniture, Style, Bus stop, Air conditioner
Australia	3.97	Food, wifi, Supermarket, Area, Pillow
Netherlands	3.94	Toilet, Hotel front, Coffee, Hotel breakfast
Japan	3.86	Bath, Bed, Room shower, Sightseeing

Table 3: Average ratings given by each intent group/nationality and corresponding distinguishing words

We ran the following experiments with our implemented system. We issued numerous pairs of queries, one with nationality specified and one without. (Similarly for bias.) For each query in the pair, we recorded the top 10 (or top 20) hotels recommended and then compared the two lists. Let S_1 denote the list of top-10 hotels for the query without nationality, and S_2 denote the top-10 list for the same query with a nationality specified. We quantify the difference between these two lists using the Jacard distance:

$$Diff(S_1, S_2) = \frac{(S_1 \bigcup S_2) - (S1 \bigcap S2)}{|S_1| + |S_2|} \quad (13)$$

Nationality The queries in the experiment are constructed as follows: for each city (4 options), and for each user's travel intent (5 options), we randomly select five different user preference values for each aspect. We calculate the distance $Diff(S_1, S_2)$ for each pair in the formed lists, and compute the average distance across all pairs of queries. This captures the average influence on the search results of including nationality. The total number of queries executed was 2500. We found that the nationality parameter affects 16.6% of the search results, thus, we believe that this piece of context is important to include in our method. We executed the same experiment with 20 hotels in the results sets, and found similar results; in this case we observed that the nationality context affected 15% of hotels recommended.

Bias adjustment. Similarly, we may wonder to what extent the bias adjustment plays a role in affecting the order of hotels presented to a user. Hence, we ran similar experiments. For each city, for each user travel intent, and for each nationality, we randomly select five different user preference values for each aspect. We compare the two obtained sets using our $Diff(S_1.S_2)$ metric. We ran 2500 such experiments. Since the bias parameter is intended to be used as a sort of tiebreaker, to differentiate very closely ranked hotels, we don't expect it to have a large impact; however if it plays no role then it could be eliminated from our method. We found that the bias affected 9% of the search results. Executing the same experiment, but recording the top-20 recommended hotels after each query, we found similar results - the bias influenced 8% of the recommendations. We believe this is a sufficiently influence to warrant retaining the bias parameter in our solution.

6. SYSTEM EVALUATION

Evaluation of a recommender system has to measure whether real people are willing to act based on the recommendations. User satisfaction with a recommender system results is well gauged with an on-line evaluation methodology. We use such a methodology as described in [30]; this methodology doesn't measure absolute user satisfaction but only relative user satisfaction with one system over another.

We implemented our system and made it available on the public web for use. We asked numerous friends and colleagues to evaluate our system and obtained 150 evaluations. Each experiment consisted of the following. The user inserts her search parameters: intent, nationality, aspect preferences, and a price range. Then we present the user a list of six hotels. Some are from our system, and some are the highest star ratings choices from Venere and Tripadvisor. In order to avoid biasing the user, these six hotels are presented in random order and thus the user is unaware of the source of the recommendations. Raters were shown links to the full text reviews to further explore the recommended hotels. For each one of the hotels, raters were asked to express their satisfaction by answering the question "Would you select this hotel?" with three optional answers: Yes, Maybe, No. In addition, the raters were asked to rate all the recommended hotels on the scale of $[1-5]$ to indicate whether they felt the recommendation had met their search criteria and was to their satisfaction. They were also asked to indicate which aspect was the one that most influenced their decision. Raters were specifically instructed to only select 'intents' that were realistic for them (e.g., if you don't have kids, do not select the 'family' as the intent).

First we look at the overall satisfaction, namely the user's response to the question "would you stay in this hotel?". We averaged the responses over all raters. We see that for 60.2% of the hotels recommended by our system, users stated they would stay there, as compared to 50.8% from the rating systems. Moreover, users stated they would not stay in 26.4% of hotels recommended by the rating systems, compared with a much lower 15.9% dissatisfaction with hotels recommended by our system. To examine this in more detail than just averages, we plot the empirical histogram of the ratings given by our raters in Figure 2. The hotels recommended by our method received more 4 and 5 ratings then those the other method, and similarly our recommended hotels received fewer 1 and 2 ratings than the other method.

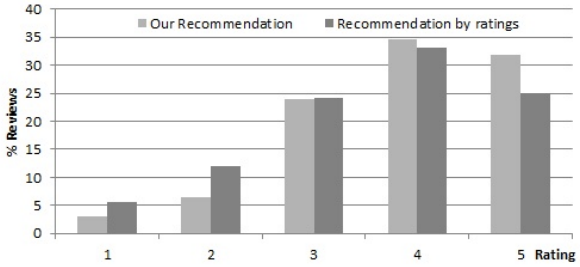

Figure 2: Ratings by recommender method

Interestingly, we observed that the satisfaction/dissatisfaction results varied by country (Germany versus Italy in our data), as can be seen in Table 6. The Italian hotels have higher average price as well as higher price variance than those in Germany. This makes producing a good recommendation in Italy harder, because when the price is reasonable (as in our German hotels) people are more easily satisfied. Satisfaction with the star rating recommendations in Italy was at 47.4%, whereas for our context-based system it rose to 58.8%. Similarly, our system has a more dramatic affect in terms of lowering dissatisfaction for the Italian hotels (dissatisfaction is lowered from 30.1% to 15.4%), than for the German ones.

An important issue in understanding the performance of a cold-start context-based recommender is to assess user consistency. We looked at the reason each rater stated for making their decisions (to stay or not, and their rating of our proposals). We compared that to their preference markings for hotel aspects. In Table 6, we show that our users are indeed very consistent; their decisions were consistent $78\% - 97\%$ of the time (depending upon the case). For the case where users made consistent decisions, we show in $63 - 72\%$ those cases, the user marked that they were satisfied. This indicates that we showed them hotels whose reviews resonated positively for them because the focused on the aspects user care about and make decisions on.

City	System	Price Avg, Stdv	Yes	Maybe	No
Germany	Ours	83, 40	62.2%	21.1%	16.7%
Germany	Stars	83, 40	55.6%	23.0%	21.4%
Italy	Ours	99, 70	58.8%	25.8%	15.4%
Italy	Stars	99, 70	47.4%	22.5%	30.1%

Table 4: User satisfaction by destination country

Type / Reason	Location	Service	Room	Price	Food
Consistent	97.1%	81.6%	88.5%	78.9%	84.4%
Satisfy	65.9%	68.8%	65.7%	72.6%	63.2%

Table 5: Users reason consistency

Next, we checked whether our recommendations resonate well for users with different trip intents. Table 6 details users satisfaction by intent. Satisfaction from our results was higher than for the star rating sites by 13% on average. Single and business travelers were considerably more satisfied with our suggestions than by those of the star ratings systems, and showed a considerable lower dissatisfaction. Interestingly, users who planned to travel in a group were dramatically more satisfied with our system, with 21.4% preferring our systems' suggestions.

Intent	Recommender System	Yes	No
Business	Our System	60.7%	16.0%
	Star Rating	50.0%	25.0%
Single	Our System	65.5%	15.5%
	Star Rating	52.8%	26.4%
Couple	Our System	55.7%	16.3%
	Star Rating	53.7%	22.6%
Family	Our System	52.7%	13.8%
	Star Rating	44.4%	36.1%
Group	Our System	66.6%	17.2%
	Star Rating	45.2%	28.5%

Table 6: Satisfy by intent

7. CONCLUSIONS

We have demonstrated that common traits for groups can be found by preprocessing large samples of text. This is a powerful result, as identifying group traits can later be used for classifying whether unknown individuals belong to the group. Additionally, if common traits of a group are known, text of reviews can be mined to identify the typical crowd of a restaurant or a hotel, for example.

Additionally, an interesting outcome of our Mechanical Turk experiments suggests that there is no strict correlation between how a review is perceived and the corresponding rating given by its author.

8. REFERENCES

[1] N. M. T. Watch, "Online Travel Market," April 2011. [Online]. Available: http://www.newmediatrendwatch.com/world-overview/91-online-travel-market

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data*, 2005.

[3] A. Rashid, I. Albert, D. Cosley, S. Lam, S. McNee, J. Konstan, and J. Riedl, "Getting to know you: learning new user preferences in recommender systems," in *Proceedings of the 7th ACM international conference on Intelligent user interfaces*, 2002.

[4] A. Schein, A. Popescul, L. Ungar, and D. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th annual international ACM SIGIR conference*, 2002.

[5] W. Woerndl and J. Schlichter, "Introducing context into recommender systems," in *Short Paper, Proc. AAAI 2007 Workshop on RecSys in e-Commerce*, 2007.

[6] T. Jiang and A. Tuzhilin, "Improving personalization solutions through optimal segmentation of customer bases," *IEEE transactions on knowledge and data engineering*, 2008.

[7] N. Hariri, Y. Zheng, B. Mobasher, and R. Burke, "Context-aware recommendation based on review mining," *General Co-Chairs*, 2011.

[8] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," *Recommender Systems Handbook*, 2011.

[9] Y. Park and A. Tuzhilin, "The long tail of recommender systems and how to leverage it," in *Proceedings of the 2008 ACM conference on RecSys*, 2008.

[10] G. Al Mamunur Rashid and J. Riedl, "Learning preferences of new users in recommender systems: an information theoretic approach," *ACM SIGKDD Explorations Newsletter*, 2008.

[11] L. Zhang, X. Meng, J. Chen, S. Xiong, and K. Duan, "Alleviating cold-start problem by using implicit feedback," *Advanced Data Mining and Applications*, 2009.

[12] X. Lam, T. Vu, T. Le, and A. Duong, "Addressing cold-start problem in recommendation systems," in *Proceedings of the 2nd ACM international conference on Ubiquitous information management and communication*, 2008.

[13] S. Park and W. Chu, "Pairwise preference regression for cold-start recommendation," in *Proceedings of the third ACM conference on RecSys*, 2009.

[14] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD*. ACM, 2004.

[15] ——, "Mining opinion features in customer reviews," in *Proceedings of the National Conference on AI*. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.

[16] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, 2010.

[17] M. Girvan and M. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, 2002.

[18] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Physical Review E*, 2006.

[19] ——, "Detecting fuzzy community structures in complex networks with a potts model," *Physical Review Letters*, 2004.

[20] K. Church, W. Gale, P. Hanks, and D. Kindle, "6. using statistics in lexical analysis," *Lexical acquisition: exploiting on-line resources to build a lexicon*, 1991.

[21] B. Liu, "Sentiment analysis and subjectivity," *Handbook of Natural Language Processing,*, 2010.

[22] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, 2008.

[23] X. Ding, B. Liu, and P. Yu, "A holistic lexicon-based approach to opinion mining," in *Proceedings of the international conference on Web search and web data mining*. ACM, 2008.

[24] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Introduction to wordnet: An on-line lexical database*," *International Journal of lexicography*, 1990.

[25] S. Brody and N. Elhadad, "An unsupervised aspect-sentiment model for online reviews," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter*. Association for Computational Linguistics, 2010.

[26] S. Aciar, D. Zhang, S. Simoff, and J. Debenham, "Informed recommender: Basing recommendations on consumer product reviews," *Intelligent Systems, IEEE*, 2007.

[27] N. Jakob, S. Weber, M. Müller, and I. Gurevych, "Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations," in *Proceeding of the 1st ACM international CIKM workshop*, 2009.

[28] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: Improving rating predictions using review text content," in *12th International Workshop on the Web and Databases*. Citeseer, 2009.

[29] S. Faridani, "Using canonical correlation analysis for generalized sentiment analysis, product recommendation and search," in *Proceedings of the fifth ACM conference on RecSys*, 2011.

[30] C. Hayes and P. Cunningham, "An on-line evaluation framework for recommender systems," 2002.