Discovering Contextual Information from User Reviews for Recommendation Purposes

Konstantin Bauman
Stern School of Business
New York University
kbauman@stern.nyu.edu

Alexander Tuzhilin
Stern School of Business
New York University
atuzhili@stern.nyu.edu

ABSTRACT

The paper presents a new method of discovering relevant contextual information from the user-generated reviews in order to provide better recommendations to the users when such reviews complement traditional ratings used in recommender systems. In particular, we classify all the user reviews into the "context rich" *specific* and "context poor" *generic* reviews and present a *word-based* and an *LDA-based* methods of extracting contextual information from the *specific* reviews. We also show empirically on the Yelp data that, collectively, these two methods extract almost all the relevant contextual information across three different applications and that they are complementary to each other: when one method misses certain contextual information, the other one extracts it from the reviews.

Keywords

Recommender systems; Contextual information;
Online reviews; User-generated content

1. INTRODUCTION

The field of Context-Aware Recommender Systems (CARS) has experienced extensive growth since the first papers on this topic appeared in the mid-2000's [3] when it was shown that the knowledge of contextual information helps to provide better recommendations in various settings and applications, including Music [8, 9, 12, 13], Movies [5], E-commerce [17], Hotels [10], Restaurants [14].

One of the fundamental issues in the CARS field is the question of what context is and how it should be specified. According to [2, 7], context-aware approaches are divided into *representational* and *interactional*. In the representational approach, adopted in most of the CARS papers, context can be described using a set of observable contextual variables that are known a priori and the structure of which does not change over time. In the interactional approach [4, 11], the contextual information is not known a priori and either needs to be learned or modeled using latent approaches, such as the ones described in [11]. Although most of the CARS literature has focused on the representational approach, an argument has been made that the context is not known in advance in many CARS applications and, therefore, needs to be discovered [3].

CBRecSys 2014, October 6, 2014, Silicon Valley, CA, USA.

In this paper, we focus on the interactional approach to CARS and assume that the contextual information is *not* known in advance and is latent. Furthermore, we focus on those applications where rating of items provided by the users are supplemented with user-generated reviews containing, the contextual information, among other things. For example, in case of Yelp, user reviews contain valuable contextual information about user experiences of interacting with Yelp businesses, such as restaurants, bars, hotels, and beauty & spas. By analyzing these reviews, we can discover various types of rich and important contextual information that can subsequently be used for providing better recommendations.

One way to discover this latent contextual information would be to provide a rigorous formal definition of context and discover it in the texts of the user-generated reviews using some formal text mining-based context identification methods. This direct approach is difficult, however, because of the complex multidimensional task of defining the unknown contextual information in a rigorous way, identifying what constitutes context and what does not in the user-generated reviews, and dealing with complexities of extracting it from the reviews using text mining methods.

Therefore, in this paper we propose the following *indirect* method for discovering relevant contextual information from the user-generated reviews. First, we observe that the contextual information is contained mainly in the *specific* reviews (those that describe specific visit of a user to an establishment, such as a restaurant) and hardly appears in the *generic* reviews (the reviews describing overall impressions about a particular establishment). Second, words or topic describing the contextual information should appear much more frequently in the specific than in the generic reviews because the latter should mostly miss such words or topics. Therefore, if we can separate the specific from the generic reviews, compare the frequencies of words or topics appearing in the specific vs. the generic reviews and select these words or topic having high frequency ratios, then they should contain most of the contextual information among them. This background work of applying the frequency-based method to identifying the important context-related words and topics paves the way to the final stage of inspecting these lists of words and topics.

In this paper, we followed this indirect approach and developed an algorithm for classifying the reviews into the "context rich" specific and "context poor" generic reviews. In additional, we present a *word-based* and an *LDA-based* methods of extracting contextual information from the specific reviews. We also show that, together, these two methods extract *almost all* the relevant contextual information across three different applications (restaurants, hotels, and beauty & spas) and that they are complementary to each other: when one method misses certain contextual information, the other one extracts it from the reviews and vice versa. Furthermore, in those few cases when these two methods fail to extract the relevant contextual information, these types of contexts turned out to be rare (appear infrequently in the reviews) and are more subtle (i.e., it is hard to describe such contexts in crisp linguistic terms).

[1, 10, 14] present some prior work on extracting contextual information from the user-generated reviews. Although presenting different approaches, these three references have one point in common: in all the three papers the types of contextual information are a priori known. Therefore, the key issue in these papers is determination of the specific values of the known contextual types based on the reviews.

Although significant progress has been made on learning context from user-generated reviews, nobody proposed any method of separating the reviews into specific and generic and presented the particular methods of extracting the contextual information from the reviews that are described in this paper.

This paper makes the following contributions. First, we proposed two novel methods, a *word-based* and an *LDA-based,* of extracting the contextual information from the user-generated reviews in those CARS applications where contexts are not known in advance. Second, we validated them on three real-life applications (Restaurants, Hotels, and Beauty & Spas) and experimentally showed that these two methods are (a) complementary to each other (whenever one misses certain contexts, the other one identifies them and vice versa) and (b) collectively, they discover almost all the contexts across the three different applications. Third, we show that most of this contextual information can be discovered quickly and effectively.

2. METHOD OF CONTEXT DISCOVERY

The key idea of the proposed method is to extract the contextual information from the user-generated reviews. However, not all the reviews contain rich contextual information. For example, *generic* reviews, describing overall impressions about a particular restaurant or a hotel, such as the one presented in Figure 1, contain only limited contextual information, if any. In contrast, the *specific* visits to a restaurant or staying in a hotel may contain rich contextual information. For example, the review presented in Figure 2 and describing the specific dining experience in a restaurant contains such contextual information as "lunch time," with whom the person went to the restaurant, and the date of the visit. Therefore, the first step in the proposed approach is to separate such *generic* from the *specific* reviews, and we present a particular separation method in Section 2.1.

After that, we use the specific/generic dichotomy to extract the contextual information using the two methods proposed in this paper, the first one based on the identification of the most important context-related words and the second one on the popular LDA method [6]. These two approaches are described in Section 2.2 and 2.3 respectively.

Figure 1: An example of a *generic* review

Figure 2: An example of a *specific* review

2.1 Separating Reviews into Specific and Generic

The main idea in separating specific from generic reviews lies in identification of certain characteristics that are prevalent in one type but not in the other type of review. For example, users who describe particular restaurant experiences tend to write long reviews and extensively use past tenses (e.g., "I came with some friends for lunch today"), while generic reviews tend to use present tense more frequently (e.g., "they make wonderful pastas").

In this work, we identified several such features for separating the generic from the specific reviews, including (a) the length of the review, (b) the total number of verbs used in the review and (c) the number of verbs used in past tenses. More specifically, we used the following measures in our study:

- *LogSentences*: logarithm of the number of sentences in the review plus one[1].
- *LogWords*: logarithm of the number of words used in the review plus one.
- *VBDsum*: logarithm of the number of verbs in the past tenses in the review plus one.
- *Vsum*: logarithm of the number of verbs in the review plus one.
- *VRatio* - the ratio of VBDsum and Vsum ($\frac{VBDsum}{Vsum}$).

Given these characteristics, we used the classical K-means clustering method to separate all the reviews into the "specific" vs. "generic" clusters. We describe the specifics of this separation method, as applied to our data, in Section 3.2.

Once the two types of reviews are separated into two different classes, we next apply the word-based and LDA-based methods described in the next two sections.

[1]We added one avoid the problem of having empty reviews when logarithm becomes $-\infty$.

2.2 Discovering Context Using Word-based Method

The key idea of this method is to identify those words (more specifically, nouns) that occur with a significantly higher frequency in the specific than in the generic reviews. As explained earlier, many contextual words describing the contextual information fit into this category. We can capture them by analyzing the dichotomy between the patterns of words in the two categories of reviews, as explained below, and identify them as follows:

1. For each review R_i, identify the set of nouns N_i appearing in it.
2. For each noun n_k, determine its weighted frequencies $w^s(n_k)$ and $w^g(n_k)$ corresponding to the specific (s) and generic (g) reviews, as follows
$$w^s(n_k) = \frac{|R_i : R_i \in specific \ and \ n_k \in N_i|}{|R_i : R_i \in specific|}$$
and
$$w^g(n_j) = \frac{|R_i : R_i \in generic \ and \ n_k \in N_i|}{|R_i : R_i \in generic|}.$$
3. Filter out the words n_k that have low overall frequency, i.e.,
$$w(n_k) = \frac{|R_i : n_k \in N_i|}{|R_i : R_i \in generic \ or \ R_i \in specific|} < \alpha,$$
where α is a threshold value for the application (e.g., $\alpha = 0.005$).
4. For each noun n_k determine ratio of its specific and generic weighted frequencies: $ratio(n_k) = \frac{w^s(n_k)}{w^g(n_k)}$.
5. Filter out nouns with $ratio(n_k) < \beta$ (e.g $\beta = 1.0$).
6. For each remaining noun n_k left after filtering step 5, find the set of senses $synset(n_k)$ using WordNet[2][16].
7. Combine senses into groups g_t having close meanings using WordNet taxonomy distance. Words with several distinct meanings can be represented in several distinct groups.
8. For each group g_t determine its weighted frequencies $w^s(g_t)$ and $w^g(g_t)$ through frequencies of its members as:
$$w^s(g_t) = \frac{|R_i : R_i \in specific \ and \ g_k \cap N_i \neq \emptyset|}{|R_i : R_i \in specific|}.$$
9. For each group g_t determine ratio of its specific and generic weighted frequencies as $ratio(g_t) = \frac{w^s(g_t)}{w^g(t_t)}$.
10. Sort groups by $ratio(g_t)$ in its descending order.

As a result of running this procedure, we obtain a list of groups of words that are sorted based on the metric ratio defined in Step 9 above. Furthermore, the contextual words are expected to be located high in the list (and we empirically show it in Section 4).

[2]WordNet is a large lexical database of English. Words are grouped into sets of cognitive synonyms, each expressing a distinct concept. Function $synset(word)$ returns a list of lemmas of this word that represent distinct concepts.

2.3 Discovering Context Using LDA-based Method

The key idea of this method is to generate a list of topics about an application using the well-known LDA approach [6] and identify among them those topics corresponding to the contextual information for that application. In particular, we proceed as follows:

1. Build an LDA model on the set of the specific reviews.
2. Apply this LDA model to all the user-generated reviews in order to obtain the set of topics T_i for each review R_i with probability higher than certain threshold level.
3. For each topic t_k from the generated LDA model, determine its weighted frequencies $w^s(t_k)$ and $w^g(t_k)$ corresponding to the specific (s) and generic (g) reviews, as follows
$$w^s(t_k) = \frac{|R_i : R_i \in specific \ and \ t_k \in T_i|}{|R_i : R_i \in specific|}$$
and
$$w^g(t_k) = \frac{|R_i : R_i \in generic \ and \ t_k \in T_i|}{|R_i : R_i \in generic|}.$$
4. Filter out the topics t_k that have low overall frequency, i.e.,
$$w(t_k) = \frac{|R_i : t_k \in T_i|}{|R_i : R_i \in generic \ or \ R_i \in specific|} < \alpha,$$
where α is a threshold value for the application (e.g., $\alpha = 0.005$).
5. For each topic t_k determine the ratio of its specific and generic weighted frequencies: $ratio(t_k) = \frac{w^s(t_k)}{w^g(t_k)}$.
6. Filter out topics with $ratio(t_k) < \beta$ (e.g. $\beta = 1.0$).
7. Sort the topics by $ratio(t_k)$ in the descending order.

As a result of running this procedure, we obtain a list of LDA topics that is sorted using the ratio metric defined in Step 5 above. Since the contextual information is usually related to the specific user experiences, we expect that these contextual LDA topics will appear high in the generated list, as in the case of the word-based method described in Section 2.2.

We next go through the lists of words and topics generated in Sections 2.2 and 2.3 and select the contextual information out of them. As is shown in Section 4, this contextual information is usually located high on these two lists and therefore can be easily identified and extracted from them. The specifics are further presented in Section 4. As we can see, the list generation methods described in Sections 2.2 and 2.3 lie at the core of our context extraction methodology and make the final context selection process easy.

In summary, we proposed a method of separating the reviews pertaining to the specific user experiences from the generic reviews. We also proposed two methods of generating contextual information, one is based on the LDA topics and another on generating list of words relevant to the contextual information.

In Section 3, we empirically validate our methods and will show their usefulness and complementarity in Section 4.

Category	*Restaurants*		*Hotels*		*Beauty & Spas*	
Cluster	specific	generic	specific	generic	specific	generic
Number of reviews	168	132	195	105	173	127
Number of reviews with context	146	25	127	13	103	9
% of reviews with context	87%	19%	65%	12%	59%	7%

Table 1: Specific vs. Generic Statistics

3. EXPERIMENTAL SETTINGS

To demonstrate how well our methods work in practice, we tested them on the Yelp data (www.yelp.com) that was provided for the RecSys 2013 competition. In particular, we extracted the contextual information from the reviews pertaining to restaurants, hotels and beauty & spas applications using the word-based and the LDA-based approaches. We describe the Yelp data in Section 3.1 and the specifics of our experiments in Section 3.2.

3.1 Dataset Descriptions

The Yelp dataset contains reviews of various businesses, such as restaurants, bars, hotels, shopping, real estate, beauty & spas, etc., provided by various users of Yelp describing their experiences visiting these businesses, in addition to the user-specified ratings of these businesses. These reviews were collected in the Phoenix metropolitan area (including towns of Scottsdale, Tempe and Chandler) in Arizona over the period of 6 years. For the purposes of this study, we used all the reviews in the dataset for all the 4503 restaurants (158430 reviews by 36473 users), 284 hotels (5034 reviews by 4148 users) and 764 beauty & spas (5579 reviews by 4272 users). We selected these three categories of businesses (out of 22 in total) because they contained some of the largest numbers of reviews and also differed significantly from each other.

The data about these businesses is specified with the following attributes: business ID, name, address, category of business, geolocation (longitude/latitude), number of reviews, the average rating of the reviews, and whether the business is open or not. The data about the users is specified with the following attributes: user ID, first name, number of reviews, and the average rating given by the user. Finally, the reviews are specified with the following attributes: review ID, business ID, user ID, the rating of the review, the review (textual description), and the date of the review. For instance, Figures 1 and 2 provide examples of restaurant reviews.

3.2 Applying the proposed methods

We applied our context discovery method to the three Yelp applications from Section 3.1 (Restaurants, Hotels and Beauty & Spas). As a first step, we have separated all the user-generated reviews into the specific and generic classes, as explained in Section 2.1. In order to determine how well this method works on the Yelp data, we manually labeled 300 reviews into specific vs. generic for each of the three applications used in this study (i.e., restaurants, hotels and beauty & spas - 900 reviews in total). This labeled data was used for computing performance metrics of our separation algorithm. The results of this performance evaluation are reported in Section 4.

We have also counted the number of occurrences of contextual information in generic and specific reviews. The results presented in Table 1 support our claim that specific reviews contain richer contextual information than generic reviews across all the three applications.

Second, we have applied the word-based method described in Section 2.2 to the Yelp data. Initially, we generated sets of nouns for restaurants, hotels and beauty & spas applications respectively. After we computed the weighted frequencies of nouns and filtered out infrequent and low-ratio words (having the thresholds values of $\alpha = 0.005$, $\beta = 1.0$), only 1495, 1292 and 1150 nouns were left in the word lists for restaurants, hotels and beauty & spas cases respectively. Finally, we combined the remaining words into groups, as described in Step 7, using the Wu&Palmer Similarity measure [19] with the threshold level of 0.9. As a result, we obtained 835, 755, 512 groups of similar nouns for the restaurants, hotels and beauty & spas categories.

Third, we have applied the LDA-based method described in Section 2.3 to the Yelp data. Initially, we pre-processed the reviews using the standard text analysis techniques by removing punctuation marks, stop words, high-frequency words, etc. [15]. Then we ran LDA on the three preprocessed sets of reviews with $m = 150$ topics for each of the three applications using the standard Python module *gensim*[18]. After generating these topics, we removed the most infrequent ones, as described in Step 4 of the LDA-based approach (setting the parameter $\alpha = 0.005$) and low-ratio topics (Step 6) having the parameter $\beta = 1.0$. As a result, we were left with 135, 121 and 110 topics for each of the three applications.

We describe the obtained results in the next section.

4. RESULTS

First, the results of separation of the user-generated reviews into the specific and generic classes are presented in Table 2 that has the following entries:

- *AvgSentences*: the average number of sentences in reviews from the generic or specific cluster.
- *AvgWords*: the average number of words in reviews from the cluster.
- *AvgVBDsum*: the average number of verbs in past tense in reviews from the claster.
- *AvgVsum*: the average number of verbs in reviews from the cluster.
- *AvgVRatio*: the average ratio of VBDsum and Vsum for reviews from the cluster.

Category	*Restaurants*		*Hotels*		*Beauty & Spas*	
Cluster	specific	generic	specific	generic	specific	generic
AvgSentences	9.59	5.04	10.38	5.58	9.36	4.54
AvgWords	129.42	55.97	147.81	65.48	134.5	50.88
AvgVBDsum	27.07	1.09	28.87	1.58	25.8	1.03
AvgVsum	91.54	23.93	107.43	28.88	107.22	25.65
AvgVRatio	0.43	0.02	0.40	0.06	0.38	0.03
Size	59.3%	40.7%	67.8%	32.2%	59.2%	40.8%
AvgRating	3.53	4.03	3.57	3.81	3.76	4.35
Silhouette	0.446		0.424		0.461	
Precision	0.87	0.89	0.83	0.92	0.83	0.94
Recall	0.83	0.91	0.83	0.92	0.88	0.90
Accuracy	0.89		0.88		0.90	

Table 2: Clusterization quality

- *Size*: size of the cluster in percents from the number of all reviews in the category (restaurants, hotels and beauty & spas).
- *AvgRating*: the average rating for reviews from the cluster.
- *Silhouette*: the silhouette measure of the clusterization quality (showing how separable the clusters are).
- *Precision*: the precision measure for the cluster.
- *Recall*: the recall measure for the cluster.
- *Accuracy*: the overall accuracy of clusterization with respect to the manual labeling.

As we can see from Table 2, the separation process gives us two groups of reviews that are significantly different in *all* the presented parameters. Further, this difference is observed not only in terms of the five parameters used in the k-means clustering method used to separate the generic from the specific reviews (first five rows in Table 2), but also in terms of the average rating (AvgRating) measure (that is significantly higher for the generic than for the specific reviews across all the three categories). Also, the silhouette measure is more than 0.4 for all the three categories and is as high as 0.46 for one of them, demonstrating significant separation of the two clusters. Finally, note that the Accuracy measure is around 0.9 across the three categories of reviews (with respect to the labeled reviews - see Section 3.2), which is a good performance result for separating the reviews.

We next extracted the contextual information from the specific reviews (produced in the previous step) using the word- and the LDA-based methods. As explained in Section 3.2, we obtained the sorted lists of 835, 755, 512 groups of words for restaurants, hotels and beauty & spas categories respectively using the word-based approach. We went through these three lists and identified the contextual variables among them - they are marked with the check marks in Column 4 (Word) in Tables 3, 4 and 5 (the numbers in parentheses next to them identify the first occurrences of the group of words in the sorted lists of the groups of words produced by the word-based method).

Similarly, as explained in Section 3.2, we obtained the sorted lists of 135, 121 and 110 topics for restaurants, hotels and beauty & spas categories respectively using the LDA-based approach. We also went through these three lists and identified the contextual variables among them - they are marked with the check marks in Column 5 (LDA) in Tables 3, 4 and 5 (the numbers in parentheses next to them also identify the first occurrences of the topics in the sorted lists of the topics produced by the LDA-based method).

	Context variable	Frequency	Word	LDA
1	Company	56.3%	✓(1)	✓(6)
2	Time of the day	34.8%	✓(77)	✓(21)
3	Day of the week	22.5%	✓(2)	✓(15)
4	Advice	10.7%	✓(13)	✓(16)
5	Prior Visits	10.2%	X	✓(26)
6	Came by car	7.8%	✓(267)	✓(78)
7	Compliments	4.9%	✓(500)	✓(74)
8	Occasion	3.9%	✓(39)	✓(19)
9	Reservation	3.0%	✓(29)	X
10	Discount	2.9%	✓(4)	X
11	Sitting outside	2.4%	X	✓(64)
12	Traveling	2.4%	X	X
13	Takeout	1.9%	✓(690)	X

Table 3: Restaurants

As Table 3 demonstrates, we identified the following types of contexts for the Restaurants category:

- *Company:* specifying with whom the user went to the restaurant (e.g., with a spouse, children, friends, coworkers, etc.).
- *Time of the day:* this context variable contains information about the time of the day, such as morning, evening and mid-day.
- *Day of the week:* specifying the day of the week (Monday, Tuesday, etc.).
- *Advice:* specifying the type of an advice given to the user, such as a recommendation from a friend or a review on Yelp. This context indicates that the user knows the opinions of other parties about the restaurant before going there.
- *Prior Visits:* specifying if the user is the first time visitor or a regular in the restaurant.

- *Came by car:* specifying if the user came to the restaurant by car or not.
- *Compliments:* specifying any types of discounts or special offers that user recieved during his visit, such as happy hour, free appetizer, special offer etc.
- *Occasion:* specifying the special occasion for going to the restaurant, such as birthday, date, wedding, anniversary, business meeting, etc.
- *Reservation:* specifying if the user made a prior reservation in the restaurant or not.
- *Discount:* specifying if the user used any types of discount deals that he or she obtained before coming to the restaurant, such as groupon/coupon, a voucher and a gift certificate.
- *Sitting outside:* specifying if the user was sitting outside (vs. inside) the restaurant during his visit.
- *Takeout:* specifying if the user did not stay in the restaurant but ordered a takeout.

Note that some of this contextual information was found using either the word-based (Company, TimeOfTheDay, DayOfTheWeek, Advice, CameByCar, Compliments, Occasion, Reservation, Discount and Takeout) or the LDA-based method (Company, TimeOfTheDay, DayOfTheWeek, Advice, PriorVisits, CameByCar, Compliments, Occasion and SitOutside).

To validate the context extraction process, we went through the 400 restaurant reviews (produced as described in Section 3.2) and identified by inspection the contextual information in these reviews. This allowed us to identify the contextual information that served as the "ground truth". Table 3 contains all the contextual information that we have found in these 400 reviews (13 different types). Note that the word- and the LDA-based methods collectively found *all* this contextual information, except for the Traveling context (that determines if the user visited the restaurant while on a travel trip in the city or that he/she lives in that city) - 12 different types of context (out of 13).

Furthermore, column 3 in Table 3 presents the frequencies with which particular types of contextual variables appear in the specific reviews of restaurants. Note that the most frequently occurring popular contexts are discovered by both the word- and the LDA-based methods. The differences between the two methods come in discoveries of less frequent contexts. It is interesting to observe that the PriorVisits context was discovered by the LDA but not by the word-based method. This is the case because this context is usually represented by such expressions as "first time," "second time," "twice" and so on, which are hard to capture by the word-based method because none of these expressions contain a clearly defined "strong" noun capturing this context. In contrast, the LDA-based approach captured this context because LDA managed to combine the aforementioned expressions into one topic.

On the other hand, such contexts as Reservation, Discount and Takeout were captured well by the word-based method since all the three contexts have clearly defined nouns characterizing these contexts (e.g., "reservation," "groupon" and "takeout" respectively). In contrast, the LDA-based method did not capture them because these words ("reservation," "groupon" and "takeout") got lost among some other irrelevant topics.

	Context variable	Frequency	Word	LDA
1	Company	37.3%	✓(4)	✓(11)
2	Occasion	24.3%	✓(1)	✓(6)
3	Reservation	12.9%	✓(18)	X
4	Time of the year	12.4%	✓(94)	✓(30)
5	Came by car	9.4%	✓(381)	✓(65)
6	Day of the week	7.4%	✓(207)	✓(41)
7	Airplane	4.9%	✓(57)	✓(40)
8	Discount	4.4%	✓(23)	X
9	Prior Visits	3.7%	X	✓(57)
10	City Event	3.4%	X	X
11	Advice	1.9%	✓(134)	✓(31)

Table 4: Hotels

	Context variable	Frequency	Word	LDA
1	Company	30.1%	✓(47)	✓(22)
2	Day of the week	18.9%	✓(8)	X
3	Prior Visits	15.2%	X	✓(25)
4	Time of the day	13.2%	✓(3)	✓(4)
5	Occasion	9.6%	✓(15)	✓(29)
6	Reservation	9.4%	✓(167)	✓(1)
7	Discount	9.2%	✓(46)	✓(39)
8	Advice	4.1%	✓(2)	✓(8)
9	Stay vs Visit	3.1%	X	✓(19)
10	Came by car	1.8%	✓(113)	✓(75)

Table 5: Beauty & Spas

Finally, nether method has discovered the Traveling context because it (a) is very infrequent and (b) is described in more subtle ways, making it difficult to capture it.

In addition to Restaurants, we have also examined the Hotels and the Beauty & Spas categories. The results are presented in Tables 4 and 5 with 10 types of contexts being discovered for the Hotels case and 10 types for the Beauty & Spas categories. Also, both methods missed the CityEvent context (an event happening in the city which is the cause of traveling to that city and staying in the hotel) for the Hotels and captured all the contextual information for the Beaty & Spas application.

As these tables demonstrate, the word- and the LDA-based methods are complementary to each other: some contexts were discovered by one but not by the other method. Further, collectively, these two methods discover most of the contextual information across the three applications examined in this paper.

Figure 3 presents the performance of the word-based discovery method across the three applications (restaurants, hotels and beauty& spas). On X-axis are the ordinal numbers of the groups of words in the word-based list produced as described in Section 3.2. On the Y-axis are the cumulative number of contexts $y(x)$ discovered by examining the first x groups of words on the list. Each line in Figure 3 corresponds to the appropriate application. The jumps on the curves correspond to the number of the first occurrence of the next contextual variable in the list of groups of words. As we can see from Figure 3, word-based method identified eight contextual variables for each application within the

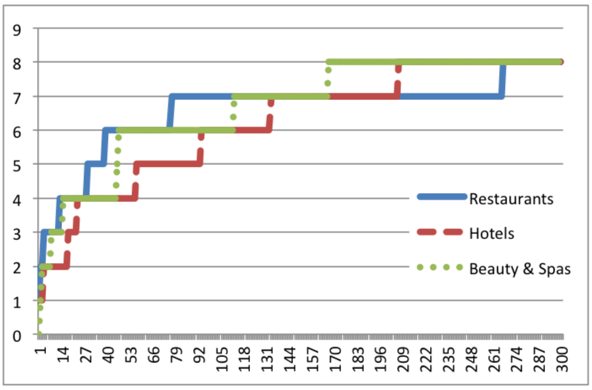

Figure 3: Word-based method

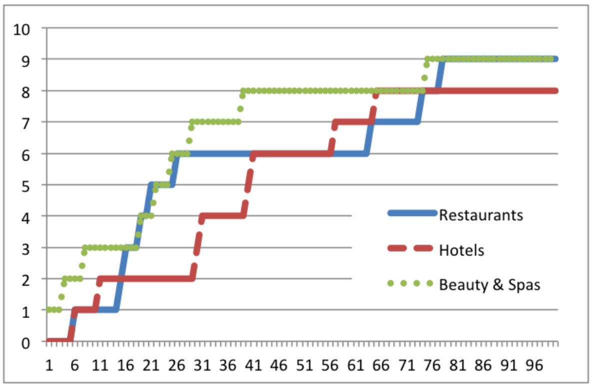

Figure 4: LDA-based method

first 300 groups of words on the list. Moreover, the first four contextual variables were identified from only first 30 groups of words on the list. This supports our earlier observation that many contextual variables appear relatively high on the list of words groups and therefore could be easily identified.

Figure 4 presents similar curves for the LDA-based method. This method managed to identify 9 contextual variables for restaurants and hotels applications, and 8 contextual variables for the beauty & spas application from the first 78 topics on the list of all the topics. Moreover, the first 6 topics were identified within just the first 41 topics. This further supports the earlier observation that many contextual variables appear high on the topics list and therefore could be easily identified.

As discussed before, the word- and the LDA-based methods are complementary to each other. In our three applications all the identified contextual variables could be identified within the first 78 LDA-topics and 29 groups of words in case of restaurants, 65 topics and 23 groups of words in case of hotels, and 75 topics and 8 groups of words in case of beauty & spas. Therefore, combination of the word- and the LDA-based methods idetifies almost all the frequent contextual variables by examining only the top several items on the two lists.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented two novel methods for systematically discovering contextual information from user-generated reviews. The first *word-based* method identifies the most important nouns that appear more frequently in the specific than in the generic reviews, and many important contextual variables appear high in this sorted list of nouns. The second *LDA-based* approach constructs a sorted list of topics generated by the popular LDA method [6]. We also show in the paper that many important types of context appear high in the list of the constructed topics. Therefore, these contexts can easily be identified by examining these two lists, as Figures 3 and 4 demonstrate.

We validated these two methods on three real-life applications (Yelp reviews of Restaurants, Hotels, and Beauty& Spas) and empirically showed that the word- and the LDA-based methods (a) are complementary to each other (whenever one misses certain contexts, the other one identifies them and vice versa) and (b) collectively, they discover almost all the contexts across the three different applications. Furthermore, in those few cases when these two methods fail to extract the relevant contextual information, the missed contexts turned out to be rare (appear infrequently in the reviews) and are more subtle (i.e., it is hard to describe these contexts in crisp terms). Finally, we showed that most of the contextual information was discovered quickly and effectively across the three applications.

As a future research, we plan to use other text mining methods in addition to the word-based and the LDA-based approaches and compare their effectiveness with the two methods presented in the paper. Hopefully, these improvements will help us to discover even more subtle and low-frequency contexts. Since the proposed word-based and LDA-based methods constitute general-purpose approaches, they can be applied to a wide range of applications, and we plan to test them on various other (non-Yelp based) cases to demonstrate broad usefulness of these methods.

6. REFERENCES

[1] S. Aciar. Mining context information from consumers reviews. In *Proceedings of Workshop on Context-Aware Recommender System*. ACM, 2010.

[2] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin. Context-aware recommender systems. *AI Magazine*, 32(3):67–80, 2011.

[3] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 217–253. Springer US, 2011.

[4] S. Anand and B. Mobasher. Contextual recommendation. In B. Berendt, A. Hotho, D. Mladenic, and G. Semeraro, editors, *From Web to Social Web: Discovering and Deploying User and Content Profiles*, volume 4737 of *Lecture Notes in Computer Science*, pages 142–160. Springer Berlin Heidelberg, 2007.

[5] J. F. T. Ante Odic, Marko Tkalcic and A. Kosir. Predicting and detecting the relevant contextual information in a movie-recommender system. In *Interacting with Computers*, 25(1), pages 74–90. Oxford University Press, 2013.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[7] P. Dourish. What we talk about when we talk about

context. *Personal Ubiquitous Comput.*, 8(1):19–30, Feb. 2004.

[8] N. Hariri, B. Mobasher, and R. Burke. Context-aware music recommendation based on latenttopic sequential patterns. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 131–138, New York, NY, USA, 2012. ACM.

[9] N. Hariri, B. Mobasher, and R. Burke. Query-driven context aware recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 9–16, New York, NY, USA, 2013. ACM.

[10] N. Hariri, B. Mobasher, R. Burke, and Y. Zheng. Context-aware recommendation based on review mining. In *IJCAI' 11, Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (ITWP 2011)*, pages 30–36, 2011.

[11] X. Jin, Y. Zhou, and B. Mobasher. Task-oriented web user modeling for recommendation. In *Proceedings of the 10th international conference on User Modeling*, UM'05, pages 109–118, Berlin, Heidelberg, 2005. Springer-Verlag.

[12] M. Kaminskas and F. Ricci. Location-adapted music recommendation using tags. In J. Konstan, R. Conejo, J. Marzo, and N. Oliver, editors, *User Modeling, Adaption and Personalization*, volume 6787 of *Lecture Notes in Computer Science*, pages 183–194. Springer Berlin Heidelberg, 2011.

[13] J. Lee and J. Lee. Context awareness by case-based reasoning in a music recommendation system. In H. Ichikawa, W.-D. Cho, I. Satoh, and H. Youn, editors, *Ubiquitous Computing Systems*, volume 4836 of *Lecture Notes in Computer Science*, pages 45–58. Springer Berlin Heidelberg, 2007.

[14] Y. Li, J. Nie, Y. Zhang, B. Wang, B. Yan, and F. Weng. Contextual recommendation based on text mining. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 692–700, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[15] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[16] G. A. Miller. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, 38:39–41, 1995.

[17] C. Palmisano, A. Tuzhilin, and M. Gorgoglione. Using context to improve predictive modeling of customers in personalization applications. *IEEE Trans. on Knowl. and Data Eng.*, 20(11):1535–1549, Nov. 2008.

[18] R. Rehurek and P. Sojka. Software framework for topic modelling with large corpora. In *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, pages 46–50, Valletta, Malta, 2010. University of Malta.

[19] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.