ActMiner: Discovering Location-Specific Activities from Community-Authored Reviews

Sahisnu Mazumder[1], Dhaval Patel[1], and Sameep Mehta[2]

[1] Indian Institute of Technology, Roorkee, Uttarakhand, India
sahisnumazumder@gmail.com, patelfec@iitr.ac.in
[2] IBM Research Lab, Delhi, India
sameepmehta@in.ibm.com

Abstract. Location-specific community authored reviews are useful resource for discovering location-specific activities and developing various location-aware activity recommendation applications. Existing works on activity discovery have mostly utilized body-worn sensors, images or human GPS traces and discovered generalized activities that do not convey any location-specific knowledge. Moreover, many of the discovered activities are irrelevant and redundant and hence, significantly affect the performance of a location-aware activity recommender system. In this paper, we propose a three-phase Discover-Filer-Merge solution, namely `ActMiner`, to infer the location-specific relevant and non-redundant activities from community-authored reviews. The proposed solution uses Dependency-aware, Category-aware and Sense-aware approaches in three sequential phases to accomplish its objective. Experimental results on two real-world data sets show that the accuracy and correctness of `ActMiner` are better than the existing approaches.

Keywords: Activity Discovery and Recommendation, Review Mining.

1 Introduction

Movement is an integral part of human daily life. Most of the times, people visit various locations to perform some activity according to their preferences. For example, people visit a restaurant with the purpose of having food, visit a shopping mall to do shopping and so on. And whenever they feel something interesting about those locations, they share those experiences with their friends using location-based social networking (LBSN) platforms like *Yelp*, *Foursquare*, *Brightkite* in terms of location-specific reviews. As of June, 2013, Yelp has been populated by over 42 million reviews for various locations in US [1]. These reviews mainly contain information about users personal experiences about locations in the form of textual description. The location-specific reviews act as a great resource for discovering activities supported by various locations (location-specific activities) and can lead to the successful development of a location-aware activity recommender system. This kind of recommender system can help people to easily figure out the best nearest location for performing a certain activity

L. Bellatreche and M.K. Mohania (Eds.): DaWaK 2014, LNCS 8646, pp. 332–344, 2014.

without wasting time in asking people and making decision from their diverse and biased suggestions.

But, the success of such a recommender system depends on the accuracy of its knowledge base which consists of a set of location-specific activities. We can utilize an approach suggested by Dearman et. al. in [2] to infer the location-specific activities in the form of (*verb, noun/noun phrase*) pairs from community-authored reviews. However, this approach doesn't consider two important issues such as relevancy and non-redundancy of the inferred activities. For example, (*watch, cricket match*) is an *irrelevant activity* for location like "*restaurant*". Similarly, activities (*have, chicken*) and (*eat, chicken*) are *redundant* to each-other. Presence of irrelevant and redundant activities in knowledge base has an adverse effect on the performance of activity recommendation. For example, separately, redundant activities like (*have, chicken*) and (*eat, chicken*) may not be the most frequent activity performed at a given restaurant. But, if we merge them together as they are redundant, their individual frequencies get added and the resultant activity may become the most frequent activity. So, presence of irrelevant and redundant information can cause incorrect activity ranking. Moreover, irrelevant and redundant information may limit the amount of useful information pushed from the recommender system to the user in broadcast environment [3].

In this paper, we propose a three-phase Discover-Filter-Merge solution, namely `ActMiner`, to infer the location-specific activities from community-authored reviews. In the first phase, `ActMiner` uses NLP techniques to discover potential (*verb, noun/noun phrase*) pairs that represent meaningful activities. In the second phase, relevant activities are discovered from the output of the first phase, using ConceptNet [4] and category information of the location. In the last phase, the redundant activities are merged if they are associated with similar sense. The major technical contributions in developing `ActMiner` are as follows:

- ***Problem Formulation***: We formulate the problem of discovering location-specific relevant and non-redundant activities from community-authored reviews. To the best of our knowledge, `ActMiner` is the first activity discovery technique that addresses the issues of relevancy and non-redundancy of the discovered activities.
- ***Novel Techniques***: **(1)** We propose novel dependency-aware activity extraction technique for meaningful activity discovery. Experimental results show that the proposed method extracts more accurate and meaningful activities compared to those inferred by the existing approach [2]. **(2)** We introduce novel category-aware relevant activity discovery technique, where we build ***C****ategory-aware* ***C****oncept* ***H****ierarchy* (CCH) for each location using ConceptNet and category of the location. CCH contains relevant concepts for a given location and is utilized in the relevant activity discovery process. **(3)** We develop novel sense-aware approach for minimizing redundancies present in the discovered activities. We use ConceptNet and CCH to discover the activities having same hidden sense and merge them into a single activity.

– ***Real-world Experiments***: Experiments are performed on two real data sets to verify the accuracy and correctness of activities discovered by `ActMiner`. We also build a location-aware activity recommender system to evaluate the effectiveness of the proposed solution.

2 Preliminaries and Related Work

Let $L = \{L_1, L_2, L_3, ..., L_m\}$ be the set of m locations and cat_i be the list of categories for location $L_i \in L$. In our database, each location L_i is associated with a set of reviews $Rset_i = \{R_i^1, R_i^2, R_i^3, ..., R_i^n\}$, where R_i^j denotes the j^{th} review of location L_i. The review R_i^j is written in the form of textual description.

Definition 1. ***Activity.*** *An activity performed at a particular location is defined as the combination of a verb with a noun or noun phrase. An activity is meaningful if it represents some "*doing*" sense. It is represented as* $A_i^j = (verb, noun/noun\ phrase)$ *and stands for the* j^{th} *activity performed at Location* L_i. *Here, a* noun phrase *is a collocation of nouns with which a verb forms association.*

Definition 2. ***Activity Frequency.*** *The frequency of an activity* A_i^j, *denoted as* $AF(A_i^j)$, *is the number of distinct reviews that has mentioned about* A_i^j *at location* L_i.

Definition 3. ***Concept.*** *The concept associated with an activity* A_i^j *is defined as the* noun *or* noun phrase *part of an activity.*

Problem Statement. Given the categories and review sets for all m locations, i.e. $\{(Cat_i, Rset_i) \mid 1 \leq i \leq m\}$, our proposed solution discovers set of *relevant* and *non-redundant* location-specific activities $Aset_i = \{A_i^j \mid 1 \leq j \leq r\}$ for each location L_i, where A_i^j is an activity performed at location L_i.

Related Work. Majority of the research works done in the area of activity recognition in past decades, have only discovered human physical activities [5],[6] and general activities at a particular location [7], [8], [9] by analysing human body movement, gesture, GPS trajectories etc. And so, these solutions are not effective in location-specific activity discovery purpose. The approach, closest to our proposed solution and concerns about inferring location-specific activities from location-specific reviews, is paper [2] which we have considered as the base paper. The approach uses sentence tokenizer to parse the review text into its individual sentences and then employ part-of-speech tagger to identify verbs and nouns in each sentence. Next, verb-noun pairs are discovered such that noun is located within 5 words following the verb. The discovered verb-noun pairs are represented in their base form and declared as the potential activities supported by the location. Although the procedure is very simple, it is accompanied with three major limitations: **(1)** The approach doesn't discover the complete set of activity. For example, the approach doesn't generate verb-noun pair if the noun occurs before verb. **(2)** The approach has only used the "*distance*" between a verb and a noun for pair generation without considering the "*dependency between words*" which is the key factor for ensuring meaningfulness of the discovered activities. Moreover, **(3)** the approach has not addressed the issues with relevancy, and non-redundancy as highlighted in the introduction section.

3 Solution Overview

Given the categories and review sets for all m locations, `ActMiner` generate activities for each location using Discover-Filter-Merge technique (See Figure 1). The Discover-Filter-Merge technique process review set $Rset_i$ for location L_i in three sequential phase, where activities are discovered in the first phase, irrelevant activities are filtered in the second stage, and redundant activities are merged in the final phase. In the remaining sections, we explain details of each phase using $Rset_i$ for location L_i.

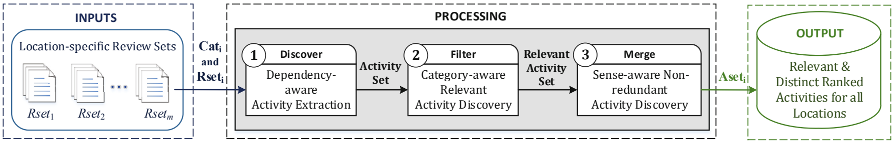


Fig. 1. Working of ActMiner

3.1 Dependency-Aware Activity Extraction

We can process each review $R \in Rset_i$ and discover any (*verb, noun*) or (*verb, noun phrase*) present in the review message. However, this simple approach generates a large number of spurious activities if the relationship between verb and noun is not taken into account. Thus, we develop a novel dependency-aware activity extraction technique that utilizes the typed dependency and proximity information between verb and noun to discover the activity. In particular, we employ a series of NLP operations on each review R in $Rset_i$, as follows:

1. We use OpennlpSentenceDetector [10] to extract individual sentences from review R.
2. Next, we parse each sentence using Stanford Typed dependency parser [11] and detect all activities in terms of (*verb, noun*) pairs that has *dobj, nsubj-pass, ccomp or prepositional grammatical relations* between verb and noun. Although dependency parser can report around 51 dependencies [12], we have observed that the aforementioned four dependencies help us to capture most the meaningful activities.
3. Dependency parser doesn't generate activity in a form of (*verb, noun phrase*). To address this issue, we extract noun phrases from each sentence using Stanford POS Tagger [13] and replace noun part of the detected (verb, noun) pair with the corresponding noun phrase part.

We explain above procedure with a help of a sample review R as shown in the Figure 2. First step extracts four sentences from R. In the next step, total five activities are discovered using dependance parsing, where (*trying, food*) activity is discovered from sentence-1 using dependency relation "*dobj*(*trying, foods*)", (*came, dinner*) is discovered from sentence-2 using "*prep_for*(*came, dinner*)", and so on. Simultaneously, POS Tagger detects *chicken tikka masala* as a noun phrase from sentence-4. Thus, pair (*enjoyed, masala*) is converted into the pair (*enjoyed, chicken tikka masala*).

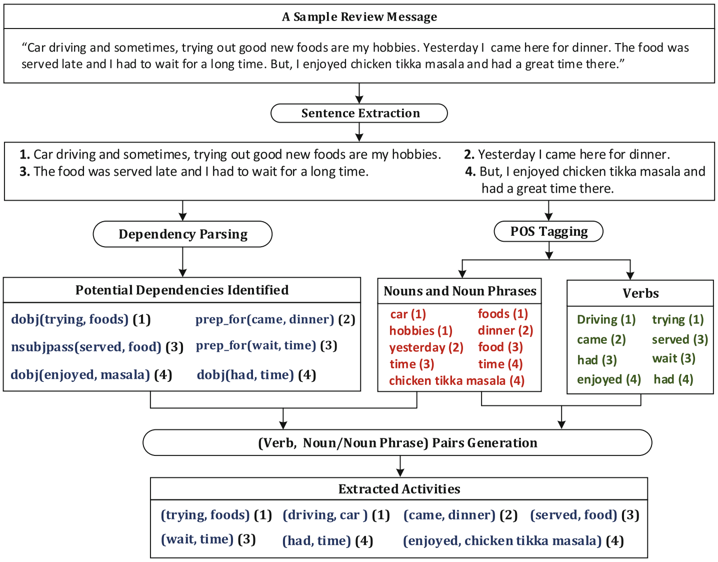


Fig. 2. An Example of Activity Extraction from Review R. The number given in "()" associated with each activity, extracted verb and noun/noun phrase indicates the sentences from which they are extracted.

Sometimes, combination of two nouns, where one noun is derived from a verb in its base form by adding "*ing*" to it, also represent a meaningful activity. In Figure 2, the word "*driving*" in sentence-1 is actually tagged as noun by pos-tagger, but it is derived from the base verb "*drive*". To deal with this case, while POS tagging a sentence, we have used WordNet [14] to check whether a noun that ends with "*ing*" has a base verb form in its synset or not. If there exists a base verb form for that noun, (for example, "*drive*" is the base verb form present in the synset of noun "*driving*"), we treat such noun as verb and generate pair by associating it with the nearest noun or noun phrase in the sentence. Thus, (*driving, car*) has been inferred as an activity from sentence-1.

Once, activities are discovered from each review R of $Rset_i$, all the verbs and nouns in the (*verb, noun/noun phrase*) pairs are converted into their base forms using WordNet. For example, (*ate, food*) is converted into (*eat, food*). Then, AF

values of each of the discovered activities are calculated and discovered activities are stored in $Aset_i$ in the form [(*verb,noun/noun phrase*), *AF*].

3.2 Category-Aware Relevant Activity Discovery

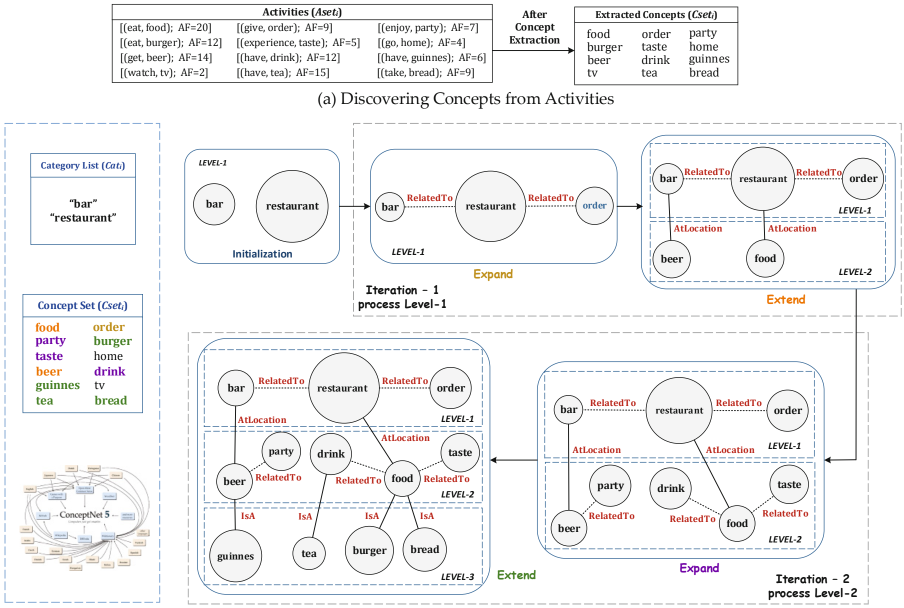
The activities discovered in the previous stage are syntactically correct, but may not represent a meaningful activity from the semantic point of view. In particular, there may be activities that cannot be performed at a given location and thus are not relevant. In this paper, we leverage ConceptNet to validate whether the concept associated with an activity conforms to the category of L_i or not. For example, given the activity (*eat, food*) at location L_i, the concept "*food*" is associated with location L_i, if category of L_i is "*restaurant*". By exploring ConceptNet, we find that concept "*food*" is related to the concept "*restaurant*" by the relation $\{food \overset{\text{At Location}}{\rightarrow} restaurant\}$. However, in most of the cases, the concept associated with an activity doesn't have any direct relationship with the category of the location. For example, concept "*chicken*" is not directly associated with the concept "*restaurant*" in ConceptNet. However, "*chicken*" is associated with "*food*" and "*food*" is intern related to "*restaurant*". Hence, a simple lookup in ConceptNet doesn't solve the the problem. However, considering the size of the ConceptNet, it is not feasible to explore all possible indirect connections. We present a systematic way of finding out the chain of relations that associates a given concept to the category of a location.

Given a set of activities $Aset_i$ for location L_i, we extract concept by processing each activity in $Aset_i$ and output the *noun or noun phrase* part of an activity as a concept. Let, $Cset_i$ be the set of discovered concepts. Then, we learn a *Category-aware Concept Hierarchy*, denoted as CCH, using $Cset_i$, category list Cat_i and ConceptNet. We use only "*IsA*", "*AtLocation*", "*DerivedFrom*", "*UsedFor*" and "*RelatedTo*" relations of ConceptNet to learn the CCH. The first three relations "*IsA*", "*AtLocation*" and "*DerivedFrom*" capture generalization-specialization relationship between two concepts. For example, $\{novel \overset{\text{IsA}}{\rightarrow} book\}$ and $\{shopper \overset{\text{DerivedFrom}}{\rightarrow} shop\}$. The remaining two relations "*UsedFor*" and "*RelatedTo*" are used for linking related concepts. For example, $\{kitchen \overset{\text{UsedFor}}{\rightarrow} cook\}$ and $\{cake \overset{\text{RelatedTo}}{\rightarrow} birthday\}$.

Definition 4. *Category-Aware Concept Hierarchy (CCH).* *A Category-aware Concept Hierarchy for a given location is a tree-based hierarchy and represented by triplet* $< Lv, C, E >$*, where* $Lv = \{lv_1, lv_2, lv_3, ..., lv_k\}$ *is the collection of levels ,* $C = \{C_1, C_2, C_3, ..., C_k\}$ *is a collection of concept sets with* C_i *be the set of concept at level* lv_i*, and* E *is the set of labeled arcs that connect concepts lying in the same level or in successive levels. The structure satisfies following 3 properties:*

- $C_1 = Cat_i$ *and* $C_i \subseteq Cset_i$ *where* $2 \leq i \leq k$.
- *If any two concepts* c *and* c' *at the same level are connected by a labeled arc* e_r*, then the label of* $e_r \in$ {*"RelatedTo", "UsedFor"*}.
- *If any two concepts* c *and* c *at level* lv_i *and* $lv_{(i+1)}$ *respectively, are connected by a labeled arc* e_r*, then label of* $e_r \in$ {*"IsA", "AtLocation", "DerivedFrom"*}.

CCH Formation. The First level of CCH is initialized with the concepts from Cat_i. Next, we perform two operations *Expand* and *Extend* iteratively to grow the hierarchy in horizontal and vertical dimensions respectively. The *Expand* operation uses "*UsedFor*" and "*RelatedTo*" relations to add concepts from $Cset_i$ into the current level. The *Extend* operation uses "*IsA*", "*AtLocation*" and "*DerivedFrom*" relations to add concepts from $Cset_i$ into the next level. In summary, the *Expand* operation adds concepts that are associated with the concepts lying in the same level and the *Extend* operation adds concepts that are specialized in sense with respect to the concepts lying in just upper level. This iterative procedure stops when the hierarchy cannot grow further.



(a) Discovering Concepts from Activities

(b) Sample Inputs

(c) Evolution of CCH in Successive Iterations

Fig. 3. CCH Formation Process

Example 1. Figure 3(a) provides a set of input activities $Aset_i$ and the set of extracted concepts $Cset_i$ from $Aset_i$. Figure 3(b) shows the sample inputs for CCH formation process which consist of $Cat_i = \{$"*restaurant*", "*bar*"$\}$, $Cset_i$ and the ConceptNet. At first, the CCH is initialized with two concepts "*restaurant*" and "*bar*" at level-1 (See Figure 3(c)). Next, the *expand* operation is performed on level-1 and concept "*order*" from $Cset_i$ is added and linked with "*restaurant*" by relation name "*RelatedTo*" at level-1. Along with that, concept "*bar*" gets associated with concept "*restaurant*" by relation "*RelatedTo*". Next, *extend* operation is performed on the expanded level-1 which adds "*beer*" and "*food*" from $Cset_i$ as a child node of "*restaurant*", links them by "*AtLocation*" relations and forms level-2. At this point *iteration-1* ends and *iteration-2* starts with execution of *expand* operation on level-2 causing expansion of level-2 by adding and linking concepts "*party*", "*drink*" and "*taste*" with existing concepts in level-2. Next, *extend* operation starts and additional concepts are added and linked with existing concepts as shown in the Figure 3(c) marking the end of *iteration-2* and begining of *iteration-3*. But in *iteration-3*, none of the concepts from $Cset_i$ gets added into the CCH and so, the CCH doesn't grow further and the process terminates.

Once construction of CCH is over, We process each activity in $Aset_i$ and declare it as relevant if the concept of activity is present in the CCH. In summary, given a set of activities, we discover concepts associated with those activities, organize them into the CCH and use this hierarchy to infer relevant activities.

3.3 Sense-Aware Non-redundant Activity Discovery

The activities discovered in the previous stage are relevant but many of them may be redundant with respect the sense of an activity. For example, both the activities (*have, food*) and (*take, food*) are associated with the same concept "*food*" and hidden sense "*eat*" and so, they are redundant to each other. To handle the activity redundancy issue, we discover the activities (having same concept) which are associated with a common hidden sense and merge them into a single activity. This process can be thought as the *sense-based activity clustering*, where each clusters is composed of a set of redundant activities. For example, considering the common hidden sense "*eat*", activities (*take, food*), (*get, food*) and (*have, food*) form a single activity cluster and is represented by (*have/get/take, food*). Similarly, example of another activity cluster is (*make/prepare, food*) formed based on the common hidden sense "*cook*". However, activities (*have, chicken*) and (*have, food*) have common hidden sense "*eat*", but as their concepts are different, they are not redundant and not merged.

Given a set of relevant activities $Aset_i$ for location L_i, we iteratively merge a pair of activities having common hidden sense and same concept. Note that, the concept of an activity is known, but activity sense is unknown. Here, we use "*RelatedTo*", "*IsA*" and "*UsedFor*" relations of ConceptNet to discover the hidden sense of an activity. For example, the common hidden sense associated with activity (*take, food*) and (*have, food*) is "*eat*" based on the relationship $\{eat \overset{\text{RelatedTo}}{\rightarrow} take\ food\}$ and $\{have\ food \overset{\text{UsedFor}}{\rightarrow} eat\}$ respectively. So, both activities are merged into single activity (*have/take, food*).

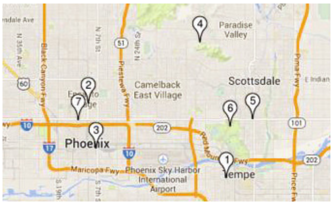
Above procedure merges activities that are mostly associated with generalized concepts such as "*food*". However, the procedure fails in most of the cases when activities are associated with specialized concepts. For example, redundant activities (*have, burger*) and (*take, burger*) are not merged using the above mentioned procedure. But, if we take the generalized concept of "*burger*" into account, we can merge (*have, burger*) and (*take, burger*) into a single activity. According to the CCH shown in Figure 3(c), the generalized concept of "*burger*" is "*food*". Since, (*have, food*) and (*take, food*) are already merged based on the common hidden sense "*eat*", we can also merge (*have, burger*) and (*take, burger*) based on the same hidden sense. This example suggests that we can apply the idea of utilizing generalized concept to merge the specialized activities. In summary, we iteratively merge two redundant activities having common concept c if the activities associated with the *generalized concept of* c are already merged. In this iterative process, we reuse CCH to infer the generalized concept.

At the end of this phase, the discovered relevant and non-redundant activities in $Aset_i$ are ranked based on their AF values and stored in a repository. Note that, merging of redundant activities increases their overall support (i.e., AF value) and helps in obtaining correct activity ranking based on their AF values.

4 Experimental Evaluation

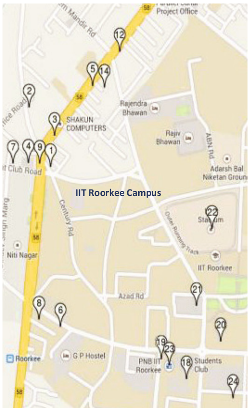
Our experiments evaluate `ActMiner` in term of its correctness, accuracy of discovered activities, and usefulness in building real world applications using two real-world review data sets, namely yelp and Roorkee. Yelp dataset contains 229,907 reviews from 43,873 users about 11,537 locations from the greater Phoenix, AZ metropolitan area. We choose 7 locations having more than 200 reviews (see Figure 4(a) and prepare a review set for each of the selected locations. Roorkee data set contains 25 locations of Roorkee that are frequently visited by many students and institute staffs from IIT-Roorkee. In total, this local data set contains 686 review messages for 25 different locations as shown in the Figure 4(b).

Loc_ID	Location Name & Address	Categories	No. of Reviews
1	**960 W University Dr** Tempe, AZ 85281, USA.	Pubs", "Bars", "Nightlife", "Restaurants"	575
2	**2611 N Central Ave**, Phoenix, AZ 85004, USA.	"Steakhouses", "Restaurants",	278
3	**401 E Jefferson St,** Phoenix, AZ 85004, USA.	"Arts & Entertainment", "Stadiums & Arenas"	216
4	**5701 N Echo Canyon Pkwy**, Phoenix, AZ 85073, USA.	"Active Life", "Climbing", "Hiking", "Parks"	210
5	**7107 E McDowell Rd**, Scottsdale, AZ 85257, USA.	"Food", "Sandwiches", "Breweries", "Pizza", "Restaurants"	232
6	**Galvin Bikeway**, Phoenix AZ 85008, USA.	"Arts & Entertainment", "Botanical Gardens", "Music Venues", "Nightlife"	260
7	**1514 N 7th Ave**, 2nd Fl, Phoenix, AZ 85007, USA.	"Bars", "Nightlife", "Lounges"	232



(a)

Loc_ID	Location Name	Categories	No. of Reviews
1	**Prakash Sweets.**	"Dessert Shop","Snacks"	61
2	**Kundan Sweets.**	"Dessert Shop","Snacks"	28
3	**Prakash Hotel.**	"Hotel", "Restaurant"	55
4	**Hotel Royal Palace.**	"Hotel", "Restaurant","bar"	38
5	**Dominos Roorkee.**	"Pizzaries"	46
6	**Sizzlers.**	"Restaurant"	13
7	**Food Point.**	"Restaurant"	19
8	**Motel Polaris.**	"Hotel", "Restaurant"	28
9	**NEEDS.**	"Convenience Store"	30
10*	**The Pentagon Mall.**	"Shopping Mall"	35
11*	**Vishal Mega Mart.**	"Shopping Mall"	22
12	**Woodland Exclusive Store.**	"Garment Shop"	18
13*	**Reebok Store.**	"Shoe Store"	16
14	**The Raymond Shop.**	"Suits", "Garments shop"	14
15*	**Nature Park.**	"Park", "Hiking"	09
16*	**Solani Park.**	"Park", "Hiking"	09
17*	**Crystal World.**	"Water Park"	16
18	**Hobbies Club.**	"club","Recreation"	26
19	**NESCAFE@IIT Roorkee.**	"Cafe","Sancks"	41
20	**Alpahar Canteen.**	"Cafe", "Snacks"	37
21	**Mahatma Ghandhi Central Library.**	"Library"	35
22	**Sports Complex.**	"Sports"	19
23	**PNB/SBI Bank.**	"Bank"	27
24	**Computer Centre.**	"Cyber cafe","Computer"	21
25*	**Railway Reservation Centre.**	"Ticket Reservation"	23

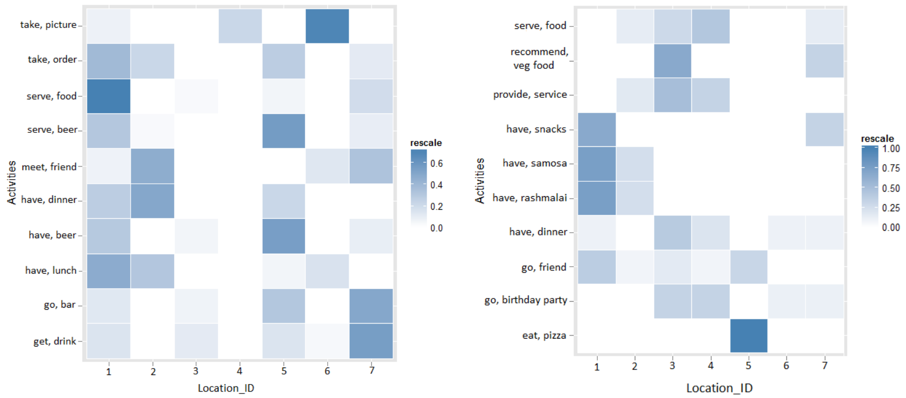


** These locations are not shown in the map due to space limitations*

(b)

Fig. 4. (a) Selected Locations from Yelp Data set. (b) Summary of Roorkee Data set.

Evaluation 1: Validating Correctness of Inferred Activities. We prepare heat maps to compare the popularity of a set of inferred activities among various locations. Figure 5 shows the heat map of activity frequency of 10 selected activities. Considering Yelp data set, (*take, picture*) is the most popular activity at location ID 6 which is a *botanical garden*. Similarly, (*serve, food*) and (*have, lunch*) activities are mostly popular in location ID 1 which is a restaurant cum bar. Considering Roorkee data set, activity (*eat, pizza*) has got the highest popularity at location ID 5 which is a pizzeria and activity (recommend, veg food)



(a) (b)

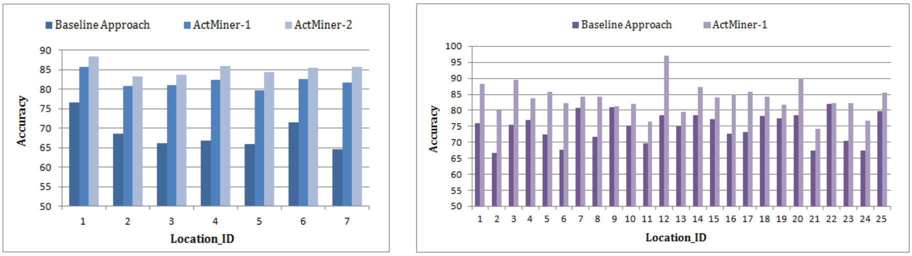
Fig. 5. Popularity of Inferred activities on (a) Yelp dataset and (b) Roorkee dataset

has got the highest popularity in location ID 3 which is the most famous hotel cum restaurant for selling vegetarian foods. In summary, we observe that the discovered information about the location-specific activities conform to the facts of the real-world scenario.

Evaluation 2: Measuring Accuracy of ActMiner. In order to measure the accuracy of discovered activities, we manually obtain the ground truth in a form of a list of activities for each location. Let, GT_i be the set of activities inferred using human perception for a location L_i and the set of activities discovered by an approach is $Aset_i$. Then, the accuracy of activity discovery for location L_i is given as $Accuracy_i = \frac{|ASet_i \cap GT_i|}{|ASet_i|}$.

For comparative study, we obtain the set of activities using baseline and two versions of `ActMiner`. `ActMiner-1` discovers activities using dependency-aware activity extraction technique, whereas `ActMiner-2` discovers activities using the idea of dependency-aware and category-aware activity discovery techniques. We have not incorporated the third, i.e., "*Merge*" phase of `ActMiner` for accuracy evaluation purpose as redundancy minimization does not effect the accuracy. On Yelp data set, we discover top 500 activities for each location using `ActMiner-1`, `ActMiner-2` and Baseline. Figure 6(a) shows that `ActMiner-1` outperforms the baseline method in terms of accuracy. More specifically, using Yelp data set, on an average, the baseline approach has achieved accuracy of 68.6% considering all 7 locations whereas dependency-aware activity extraction method, i.e., `ActMiner-1` has obtained an average accuracy of 82% showing significant improvement of 13.4%. Moreover, considering top 500 relevant activities discovered using category-aware relevant activity discovery approach, `ActMiner-2` has obtained an average accuracy of 85.23% which implies 3.23% average improvement in accuracy to that obtained using `ActMiner-1`.

On Roorkee data set, we discover all activities for each location using `ActMiner-1` and Baseline. Figure 6(b) shows the comparison of accuracies for both the approaches. Again, we observe that the activities discovered by



(a) (b)

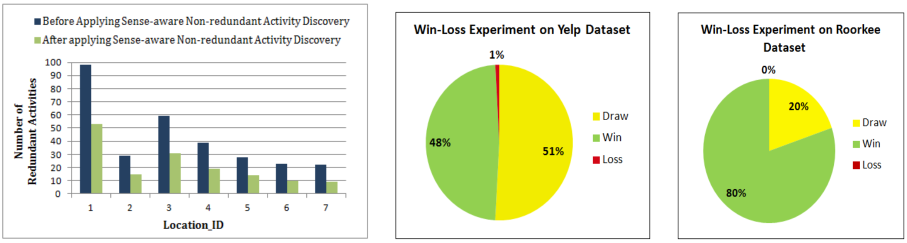
Fig. 6. Accuracy of Inferred activities on (a) Yelp dataset and (b) Roorkee dataset

`ActMiner-1` are more accurate. In terms of statistics, on an average, the baseline approach has achieved accuracy of 74.787% considering all 25 locations whereas `ActMiner-1` has obtained an average accuracy of 83.728% showing significant improvement of 8.94% in accuracy compared to the baseline approach. We have not obtained the accuracy of `ActMiner-2` on Roorkee dataset as the reviews in this dataset contain local or Indian concepts that are mostly not available in the ConceptNet and hence, are not suitable for the relevant activity detection purpose.

In summary, `ActMiner` performs more accurately than the baseline approach. The reason behind this is that we have used the dependency relations between words as the metric for activity extraction which ensures the pairing of words to be meaningful. Apart from that, most of the irrelevant activities thrown out after category-aware relevant activity discovery phase, cause improvement in accuracy in `ActMiner-2`.

Evaluation 3: Qualitative Analysis I: Broadcast Service. Now, we investigate the usefulness of sense aware redundant activity discovery phase in terms of advertising popular activities in broadcast environment. We manually obtain the list of redundant activities from the output of `ActMiner-2`. We consider these set of marked activities as the ground truth for redundancy checking purpose. Next, we run the process of sense-aware non-redundant activity discovery on the output of `ActMiner-2`. Figure 7(a) shows the number of redundant activities before and after the sense aware redundancy minimization process. On an average, for all 7 locations, we have observed 51.22% redundancy elimination done by the said process. So, in summary, we can conclude that our sense-aware redundancy elimination approach has successfully eliminated almost half of the redundancies present in the discovered activities. So, this analysis indirectly ensures that using `ActMiner`, we can reduce bandwidth wastage and push more unique information to the user while recommending in broadcast environment.

Evaluation 4: Qualitative Analysis II: Recommendation System. We have developed a location-aware activity recommendation system using the inferred activities of `ActMiner`. Given an activity A and a set of locations $Lset$, the recommender system recommends a location $L \in Lset$ such that $AF(A)$ is highest for location L. We have also developed similar recommendation system



(a) (b) (c)

Fig. 7. (a) Performance of `ActMiner` in redundancy minimization. Result of Win-Loss Experiment, on (b) Yelp data set and (c) Roorkee data set.

using the baseline approach and evaluated both the recommender systems using "*Win-Loss Experiment*". In this evaluation, if the location IDs recommended by both the recommender systems are same, we have declared the result as "*Draw*". Otherwise, the recommender system which recommends the location with higher activity frequency value, wins in the experiment. We have discovered the set of distinct activities for for all locations in each data set and used them as a query input for the evaluation.

In the Figure 7(b), we observe that, the `ActMiner`-based recommender system wins 48% cases and loses in only 1% case while competing with the recommender system formed using baseline approach, whereas in 51% cases, the results are "*Draw*". Considering Roorkee dataset, `ActMiner`-based recommender system wins 80% cases and makes a draw in 20% cases without any loss as shown in Figure 7(c). From these results, it is quite clear that `ActMiner` outperforms the baseline approach with respect to the performance recommendation and this also proves the efficacy of our proposed solution.

5 Conclusion

In this paper, we propose a Discover-Filter-Merge based technique `ActMiner` to infer the location-specific relevant and non-redundant activities from community-authored reviews. The proposed solution has successfully achieved its objective using novel Dependency-aware, Category-aware and Sense-aware approaches. Experimental analysis shows that `ActMiner` discovers location-specific activities more accurately compared to the baseline approach and proves effectiveness of the solution in providing location-aware activity recommendations.

References

1. Yelp. An Introduction to Yelp: Metrics as of June 30, 2013 (2013), `http://www.yelp.co.nz/html/pdf/Snapshot_2013_Q2_en.pdf`
2. Dearman, D., Truong, K.N.: Identifying the activities supported by locations with community-authored content. ACM UbiComp, 23–32 (2010)

3. Ku, W.S., Zimmermann, R., Wang, H.: Location-based spatial queries with data sharing in wireless broadcast environments. In: IEEE ICDE, pp. 1355–1359 (2007)
4. ConceptNet 5, http://conceptnet5.media.mit.edu/
5. Lara, O., Labrador, M.: A survey on human activity recognition using wearable sensors. IEEE Communications Surveys and Tutorials 15(3), 1192–1209 (2012)
6. Pawar, T., Chaudhuri, S., Duttagupta, S.P.: Body movement activity recognition for ambulatory cardiac monitoring. IEEE Tran. on Biomedical Eng. 54(5), 874–882 (2007)
7. Zheng, K., Shang, S., Yuan, N.J., Yang, Y.: Towards Efficient Search for Activity Trajectories. In: IEEE ICDE (2013)
8. Furletti, B., Cintia, P., Renso, C., Spinsanti, L.: Inferring human activities from GPS tracks. In: ACM SIGKDD International Workshop on Urban Computing, p. 5 (2013)
9. Zheng, V.W., Zheng, Y., Xie, X., Yang, Q.: Collaborative location and activity recommendations with GPS history data. In: ACM WWW, pp. 1029–1038 (2010)
10. Apache OpenNLP Developer Documentation, https://opennlp.apache.org/
11. The Stanford Parser: A statistical parser, http://nlp.stanford.edu/software/lex-parser.shtml
12. De Marneffe, M.C., Manning, C.D.: Stanford typed dependencies manual (2008), http://nlp.stanford.edu/software/dependenciesmanual.pdf
13. Stanford Log-linear Part-Of-Speech Tagger, http://nlp.stanford.edu/downloads/tagger.shtml
14. WordNet, http://wordnet.princeton.edu/