A Tutorial On

Location-aware Review Analytics: A Big Data Perspective

Dr. Dhaval Patel

Asst. Prof., Dept. of CSE,
IIT Roorkee

and

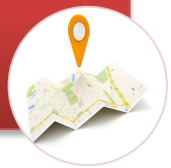
Sahisnu Mazumder

JRF, Dept. of CSE,
IIT Roorkee

Outline

- Location-aware Data and Its Collection: The First Step

Topic-I



- Location-aware Review Analytics : A Brief Introduction

Topic-II

- Location-aware Review Analytics : Research & Applications

Topic-III

- ActMiner and LANet : A Case Study of Location-aware Review Analytics

Topic-IV

- Location-aware Review Analytics: Technical Challenges

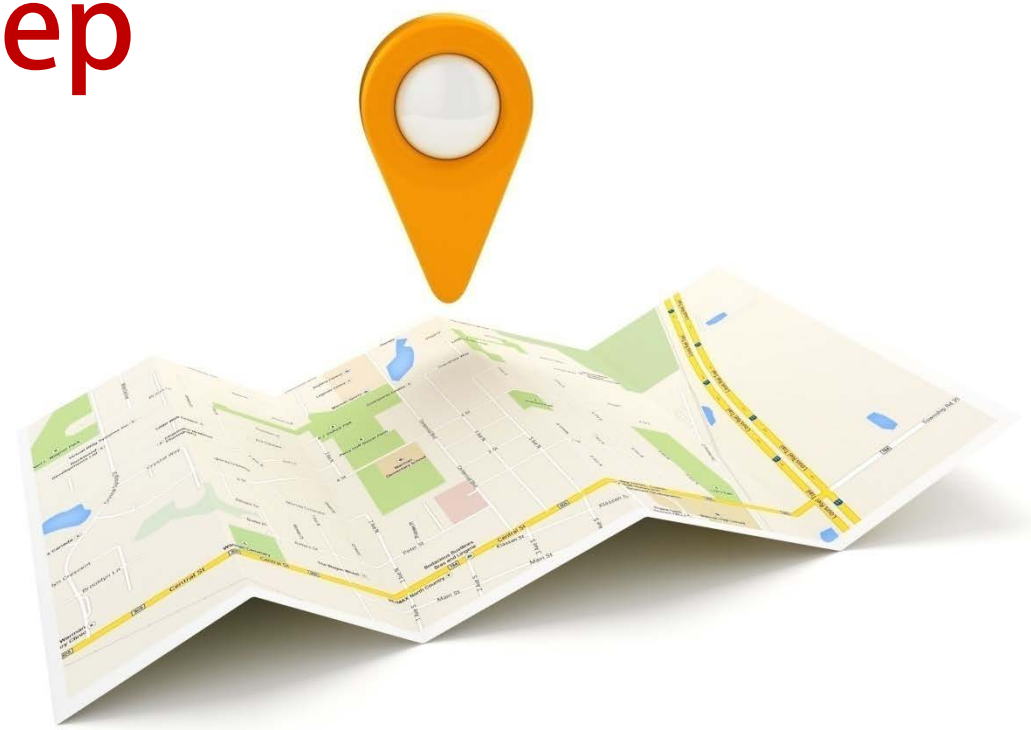
Topic-V

- NLP Techniques and Big Data Analytics Platforms : Overcoming Challenges

Topic-VI

Location-aware Data and Its Collection:

The First Step



Data comes from Everywhere

Grocery Markets

E-Commerce

Stock Exchange

Hospital

Weather Station

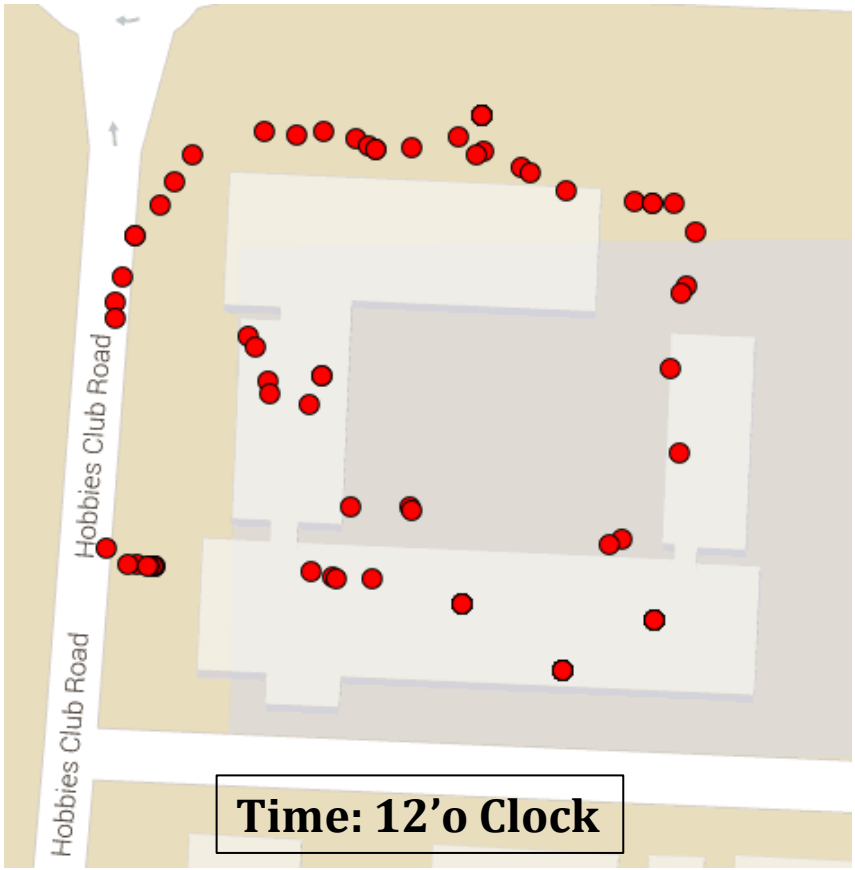
Social Media

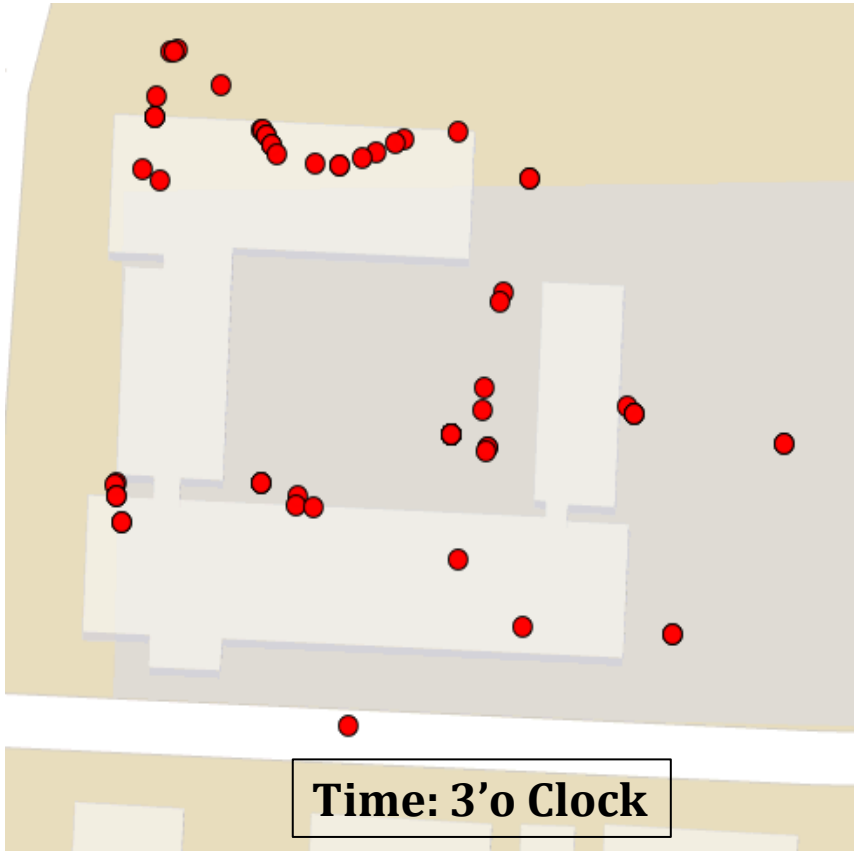
The Wireless Explosion

Do you use any of these devices ?

Do you ever feel that you (your location) are being tracked?

The Wireless Explosion: IoT Framework





Each marker is a mobile user . Their locations have been passively sensed by our framework.

Spatio-Temporal Data

- **Trajectory Data**
  - Object Movements : <{lat,long}, time> → <{lat,long}, time> → …
- **Spatio-Temporal Events**
  - What happen at **where** and **when :** <Event Id, {lat,long}, time>

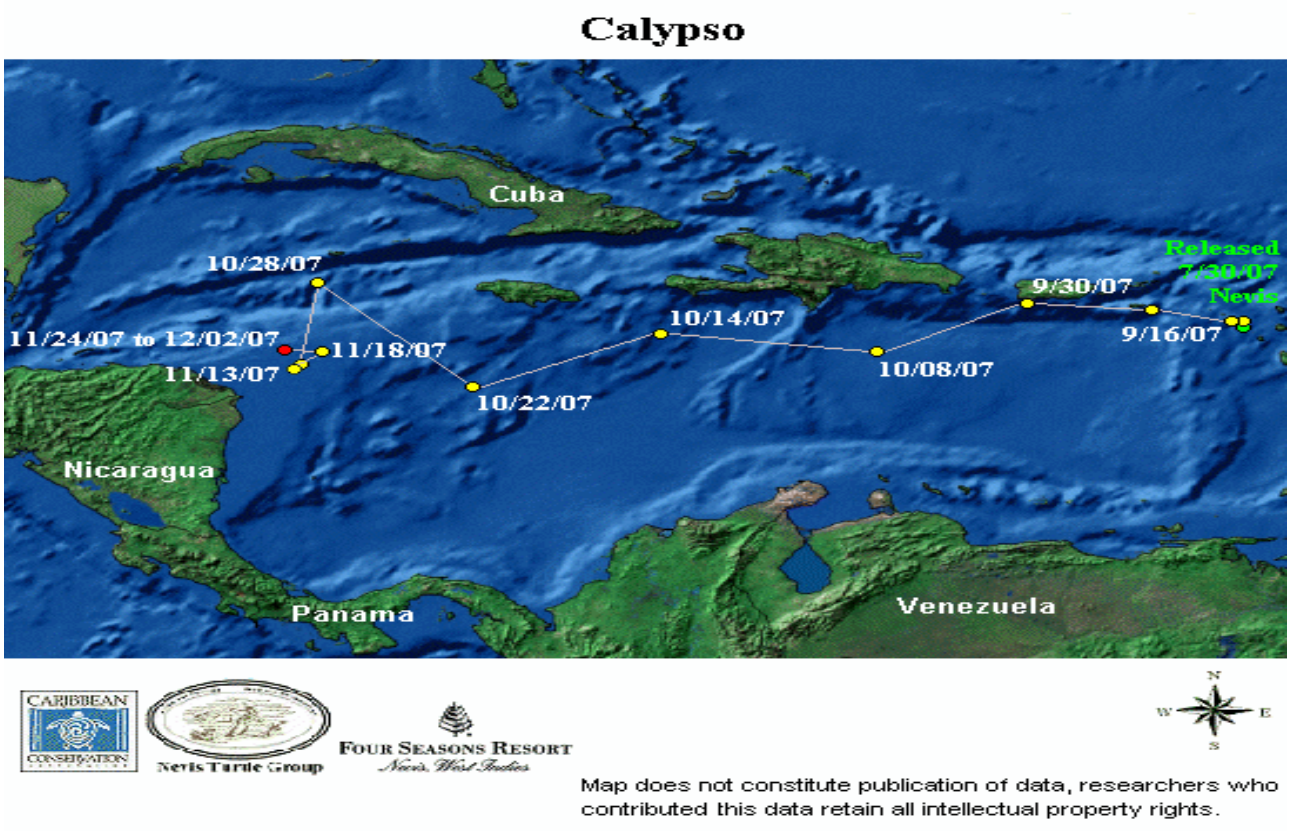
- **Location-aware Reviews**
  - User Experience:
    - What we think about a location?
    - Did we like the place or not?
  - <Location Id, Review_text/Ratings, time>
- **Digital Life Trajectory Data**
  - **location information** of the user **+ physical activities + digital activities** performed there by the user at various timestamps.
  - An integrated dataset!

→ **Emerging Data**

What are Trajectory Data?

{<25.1750, 25.2250>, 00}
{<25.2500, 25.2500>, 60}
{<25.2400, 25.2600>, 120}
...
...
{<25.2500, 25.2500>, 600}
{<25.2400, 25.2600>, 660}

A trajectory is a sequence of the location and timestamp of a moving object

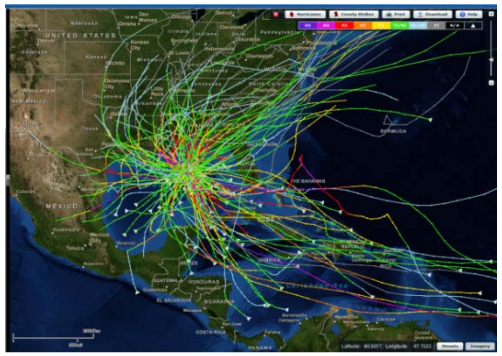


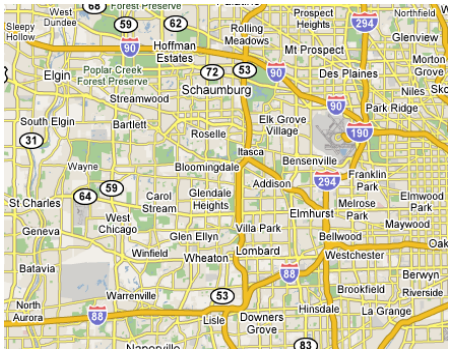
Trajectory Data are Everywhere!

We use technology to track many things…

- *Our locations*
- *Hurricanes*
- *Animal movements*
- *Public Bus Transportation Data*

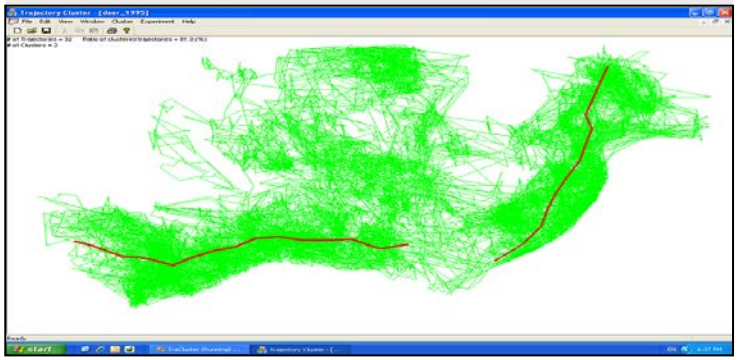
…and things' positions change over time…



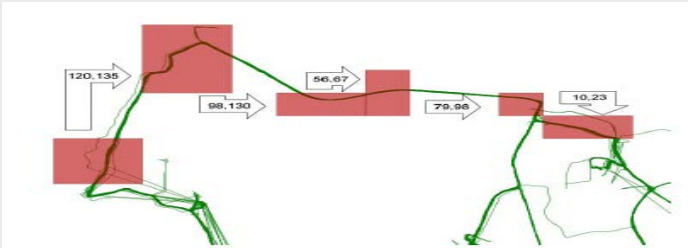


What people do with the trajectory data?

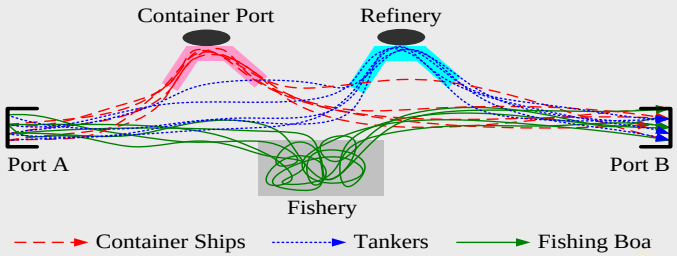
Clustering



Frequent Travel Patterns



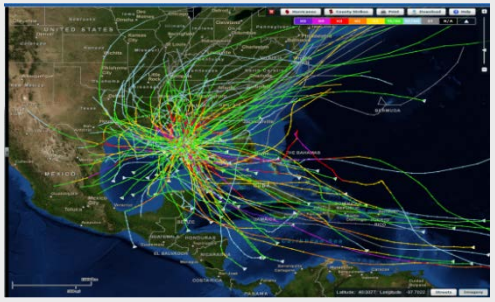
Motif Discovery



Prediction

Visualization

Classification



Spatio-Temporal Events

Meningococcal Disease in Germany

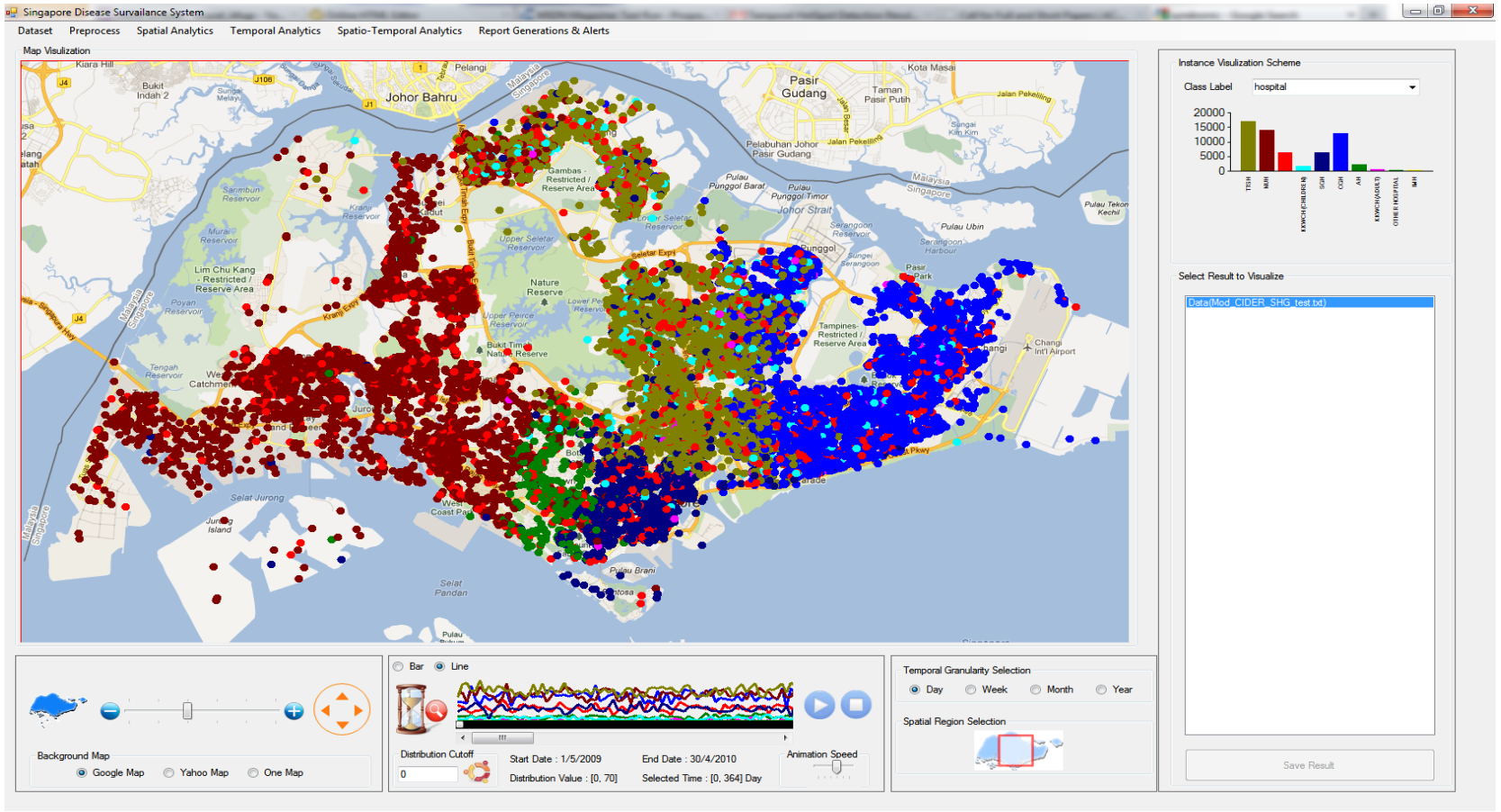
- Small **purple color dots** represent location of infected case
- **Red color dots** indicate centroids of the spatial region
- Blue color arrow suggest an infection relation between nearby spatial region
- **Brown color path** suggests spread of infectious disease

Data Domain: Infectious Disease Analysis(**EpiScanGIS-Germany**)

Application: Disease Hotspots Prediction to prevent spread of disease

Spatio-Temporal Events: An Analysis

Design of Singapore Syndromic Surveillance System



GeoVistaS

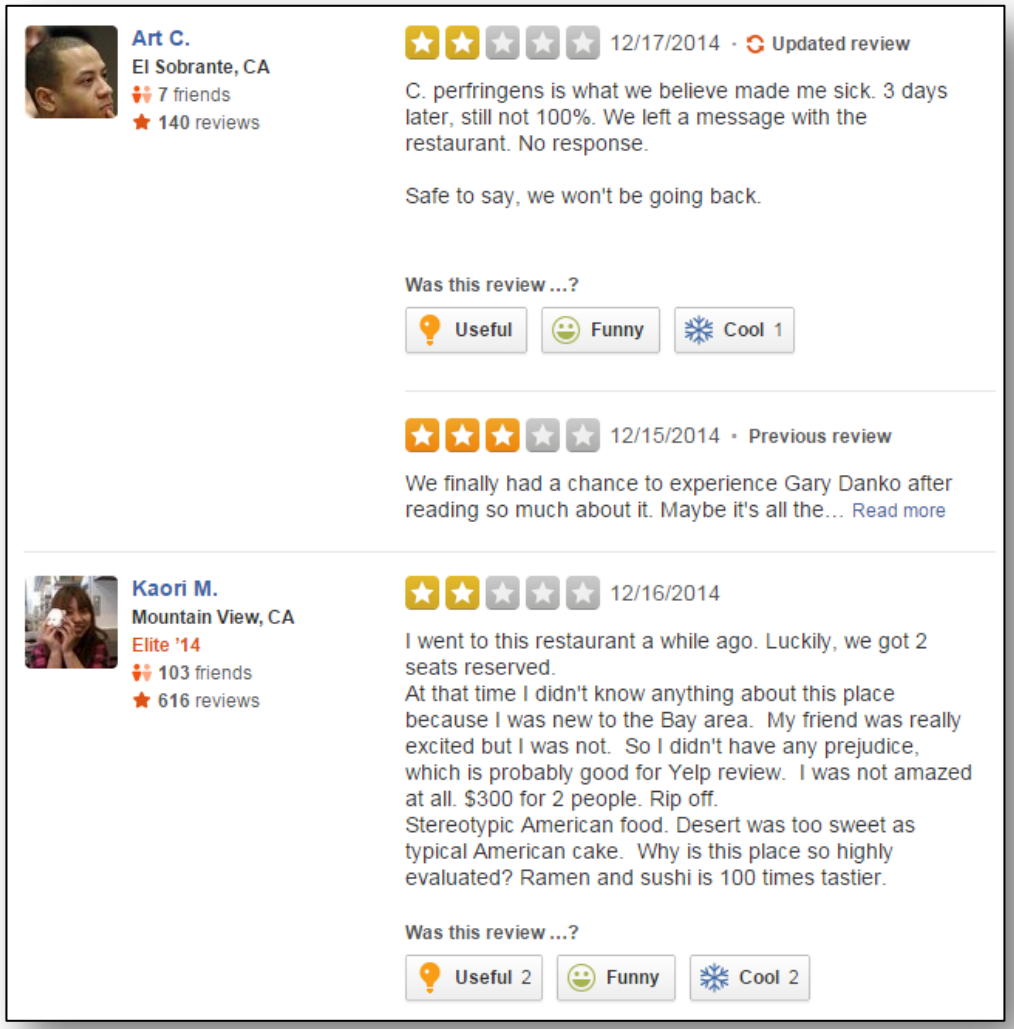
Location-aware Reviews

A textual Description and/or rating of a Location that concerns about user's visiting experience.

- Basically unstructured information.

- Contains more specific information about a location (more than its semantics) and also users' sentiments and opinions about that location.

- Can be found on **Location-based Social Networks** like yelp, foursquare and sometimes in other social media like facebook, twitter.

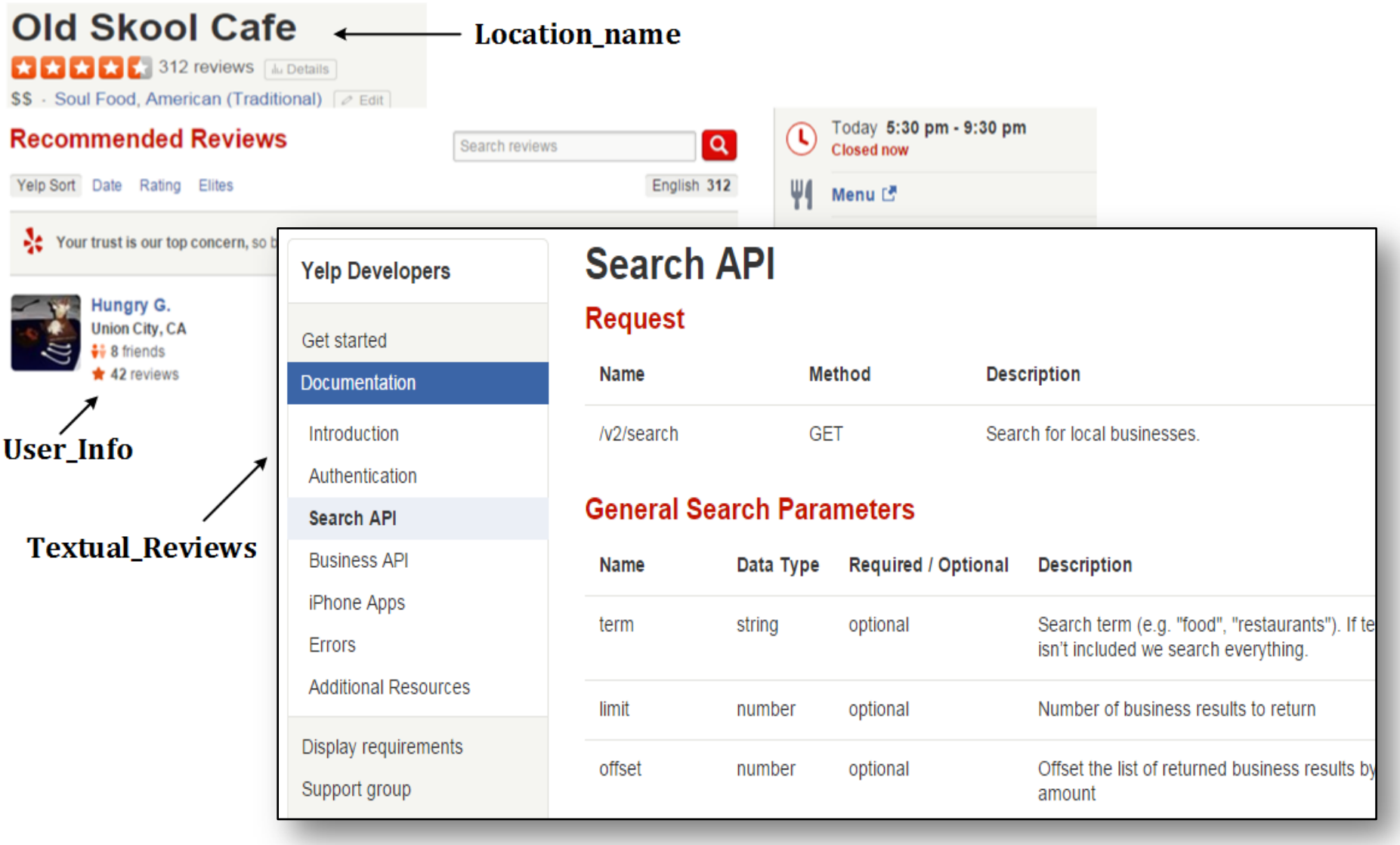


Location-based Social Networks

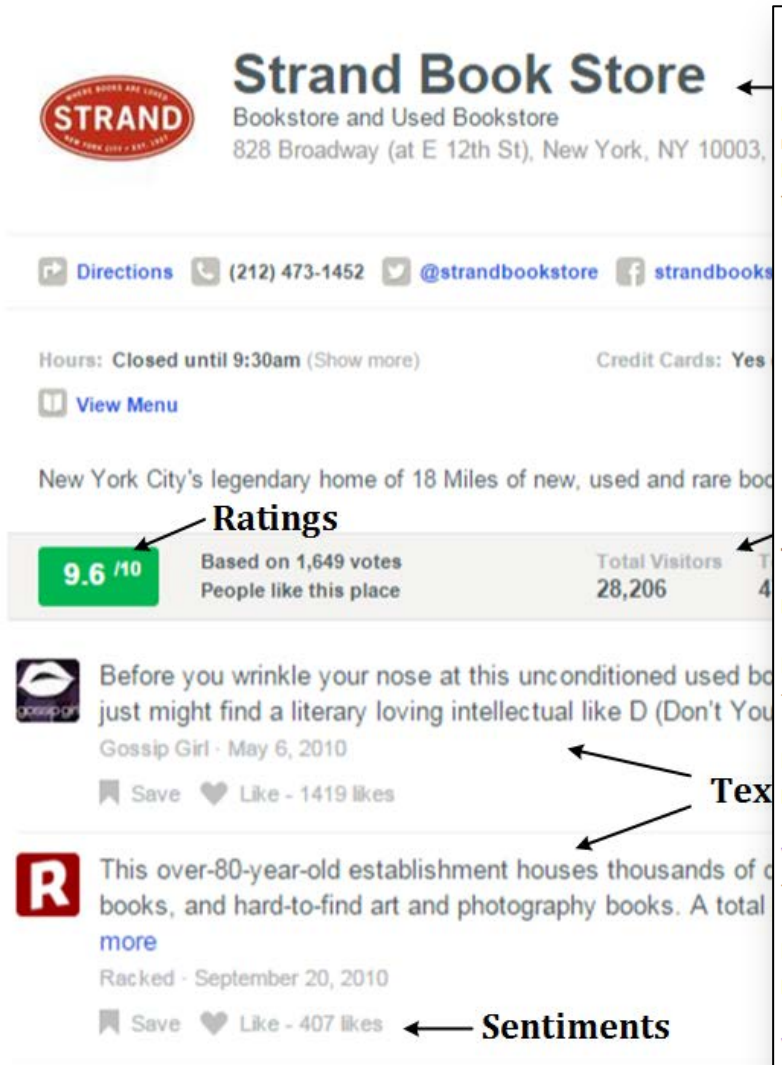
Yu Zeng,
MS Research

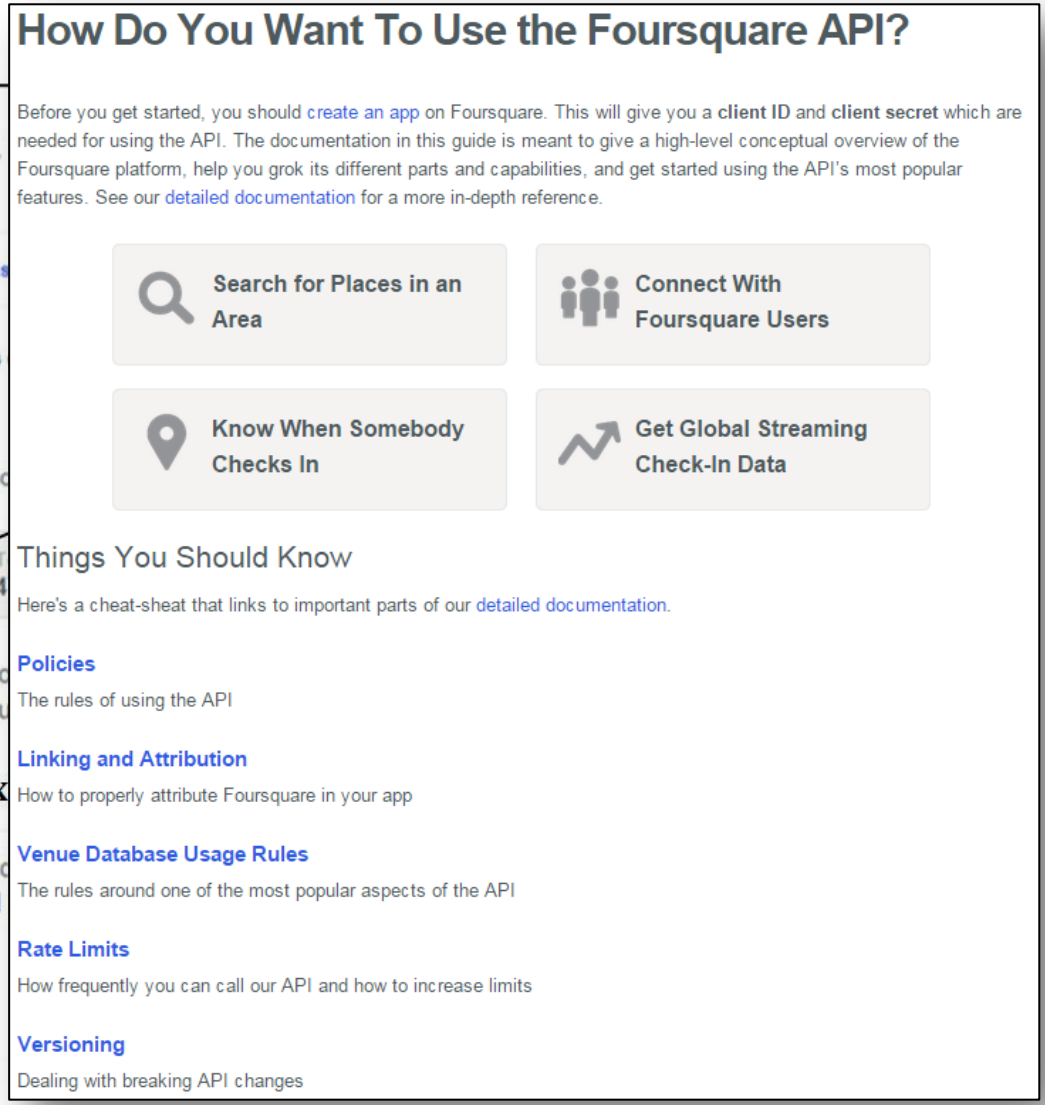
"A location-based social network (LBSN) does not only mean adding a location to an existing social network so that people in the social structure can share location-embedded information, but also ***consists of the new social structure made up of individuals connected by the interdependency derived from their locations in the physical world as well as their location-tagged media content, such as photos, video, and texts****. Here, the* ***physical location*** *consists of the* ***instant location of an individual at a given timestamp*** *and the* ***location history*** *that an individual has accumulated in a certain period. Further, the* ***interdependency*** *includes not only that* ***two persons co-occur in the same physical location or share similar location histories but also the knowledge****, e.g., common interests, behavior, and activities, inferred from an individual's location (history) and location-tagged data."*

Yelp Review Data



Foursquare Reviews and Check-in/Checkout Data

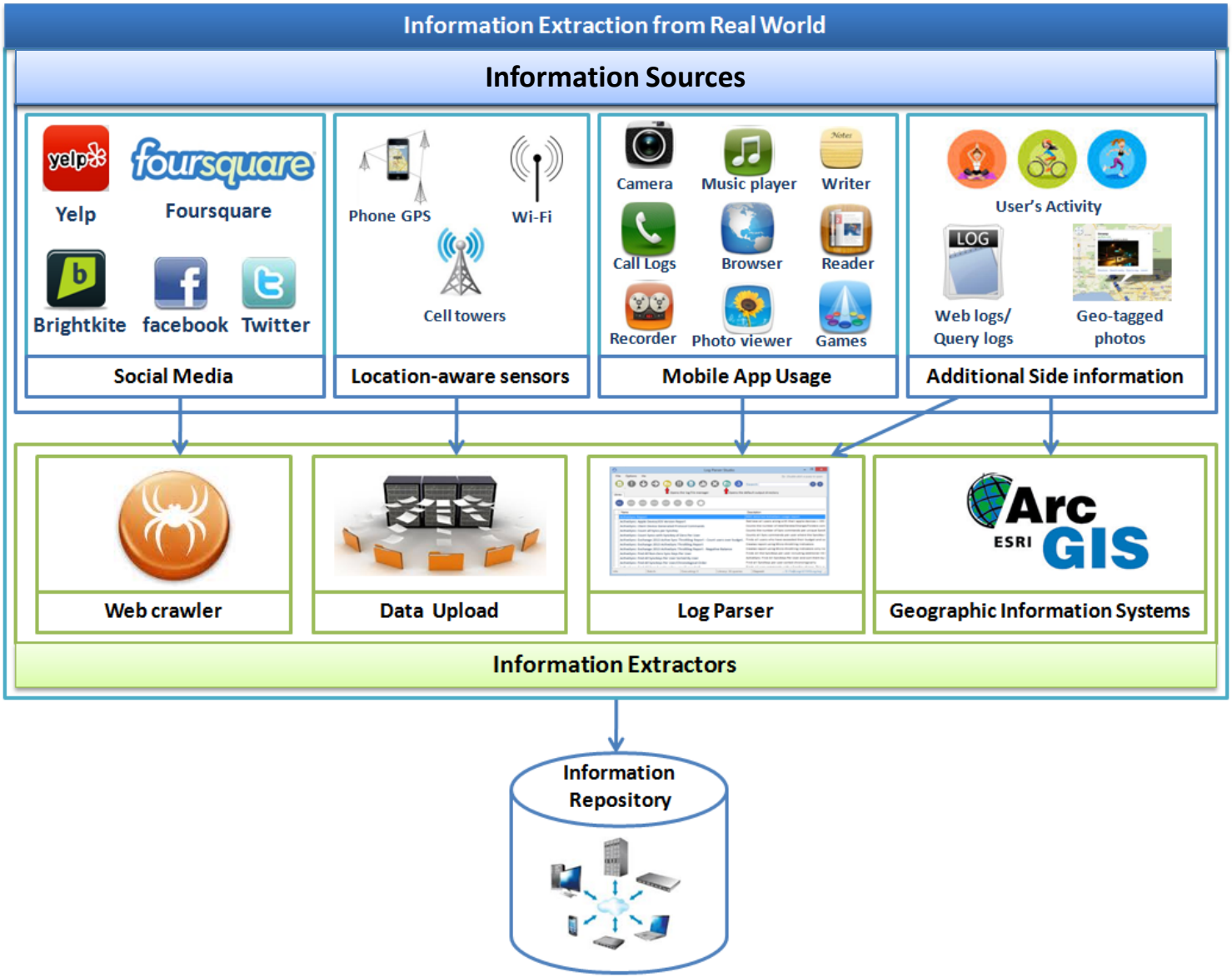




Smartphone Usage Data

Life Bookmarks of LifeLog **collect every precious detail of the events** that matter most to you; the **time of day**, your **location**, the **photos** you took and even the **weather**, ready for you to look back on and remember, forever.

In Summary, Multi-source Data Collection



Location-aware Review Analytics:

A Brief Introduction

Textual Review Analytics

- ***NOT New!***
- Product Review Analysis
  - **Aspect Discovery:** What aspects of the product do the people like?
  - **Opinion Mining/Sentiment Analysis:** Did They like product –``X" or not?
    - Hugely done by E-commerce big companies like Amazon, Flipkart etc. for product recommendation and deciding about their future business plans.

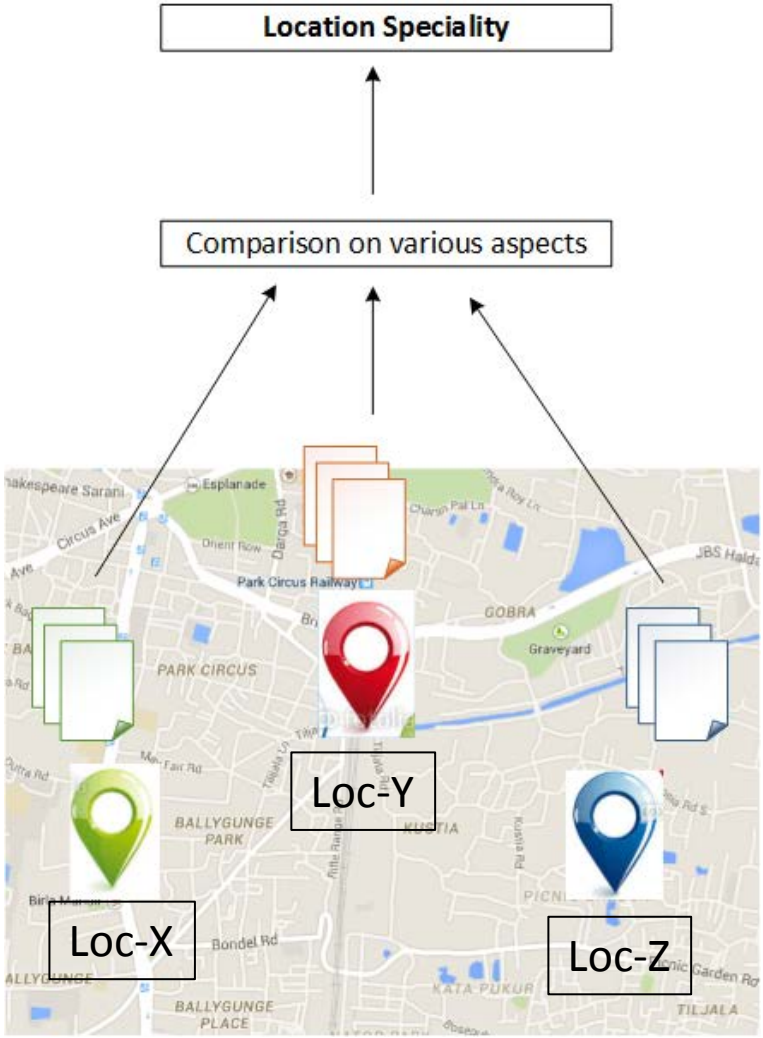
So, what's Special about the Location-aware Review Analytics?

We can compare Locations with more details !

Reviews of nearby locations can be analyzed and compared to discover interesting knowledge

- What a Location X is famous for?
- What can I do at location X?
- Should we visit a location X or not and if we should, when?
- If we do not like location X, how far we need to travel to find an alternative?
- If I own a restaurant at Loc-Y, how much my customers are satisfied?

And many more...



An Example of Knowledge Discovery from Location-aware Reviews!

Consider three popular hotel cum restaurants of Roorkee and the activities we can Perform there discovered from the reviews of the corresponding location. Here, [(have, chicken), 70%] at Loc-X denotes the activity of having chicken is 70% popular at X compared to other nearby locations.

Inference List

Hotel Center Point
1. Popular for matar paneer. 2. Best location for having dessert.

Hotel Royal Palace
1. Popular for beer. 2. Best location for having Chicken.

Alternative nearest location of Hotel Royal Palace for having chicken is Sagar Hotel & Restaurant (dist: 120m).

In summary, Location-aware Review Analytics Can Tell Us...

Location-aware Recommendation

- Nearest Hotels and restaurants from my current position.
- What activities can I perform there?

Location-aware Reviews

Location-aware Sentiment Analysis and Opinion Mining

- Is the place good?
- Should I go there?
- How many people have rated it good or bad and why?

Location -aware User Interest Discovery

- What kind of locations a person likes to visit?
- When he visits such kind of locations?
- ...

Location-aware Review Analytics: Research & Applications

Some Recent Interesting Research using Location-aware Reviews!

- How much attention a restaurant can get in future?

Inferring Future Business Attention

Bryan Hood, Victor Hwang and Jennifer King, Carnegie Mellon University

- How can a restaurant point out the demands of its customers from a large amount of reviews?

Improving Restaurants by Extracting Subtopics from Yelp Reviews

James Huang, Stephanie Rogers, Eunkwang Joo. University of California, Berkeley

- Can We make an effective visualization of user generated review that can aid us to make decisions about a location in a much quicker way?

Clustered Layout Word Cloud for User Generated Review

Ji Wang, Jian Zhao, Sheng Guo, Chris, North. Virginia Tech and University of Toronto.

Inferring Future Business Attention*

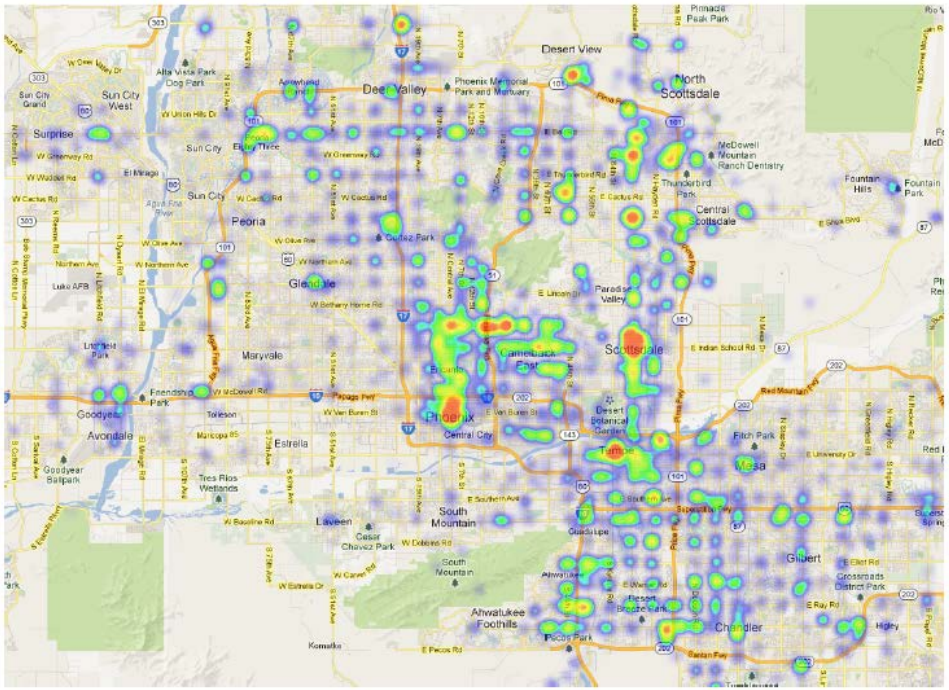
OBJECTIVE: To process and analyze Yelp user reviews, ratings, location etc. about a particular business and determine the amount of attention a business receives.

- **Idea:** Used a Yelp-provided data set to try and learn features related to a high attention business.

- **Method:**

✓ Simple manipulation of the given data and sentiment analysis on user-provided reviews.

✓ Run feature selection methods to try and pick out the best features.



Heat map of number of reviews in Phoenix, AZ

*Source: http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_InferringFuture.pdf

Feature Extraction

❑ Time-dependent features:

➢ Metadata about the business.
- Example: location (latitude and lontigude), number of businesses within 1km and number of businesses sharing a category.

➢ Features that describes a subset of reviews:
- Example: Average number of stars across reviews in the set , Maximum number of stars, Minimum number of stars, Number of reviews voted as 'cool'/ 'funny'/'useful', Number of unique users logging these reviews etc.

❑ Review Text Features:
- Mined the most frequently occurring keywords among all restaurant reviews and use counts of the sentiments for each of these keywords as features.
- **Assumption:** Restaurants with many positive statements about a particularly popular keyword would have high reviews.

*Source: http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_InferringFuture.pdf

Review Text Feature Extraction

❑ Generated a feature vector containing the counts of the number of positive and negative statements about the top 300 keywords. This is done in two steps

➢ **Review Text Parsing:** Compute the top keywords among all the reviews.
- Tokenize each review into sentences, then tokenize the sentences into words.
- Used the Python Natural Language Toolkit (NLTK) for part-of-speech tagging of each token in the sentence.
- Cared only about adjectives and nouns for phrase generation.
- Take a subset of the keywords (say, 100) that appeared among all the reviews and use this to generate feature vectors given a single review.

➢ **Keyword-Opinion Extraction:** For each review, count the number of positive and negative opinions about each keyword in order to generate a feature vector about a specific review.
- Use WordNet to generate a list of adjectives that are used to consider an opinion about a noun phrase.
- Analyze each sentence in each review by counting the number of positive and negative adjectives associated with the keywords.

*Source: http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_InferringFuture.pdf

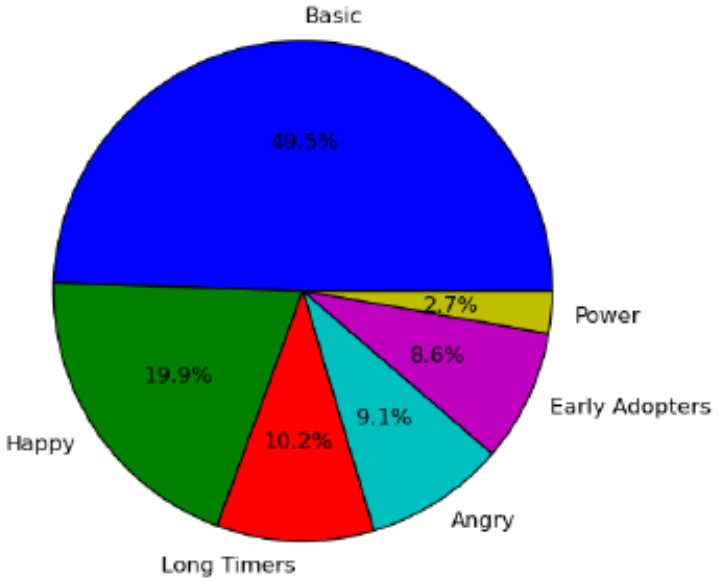
User Clustering

- For predicting the number of reviews a business has and its rating, knowing something about the user base is important.

- The number of users is quite large, so a tractable way to deal with users is to cluster them.

- Features for clustering:

(1) Number of reviews
(2) Average number of stars
(3) Number of "Funny" votes
(4) Number of "Useful" votes
(5) Number of "Cool" votes
(6) Date of first review
(7) Date of last review
(8) Days between first and last review

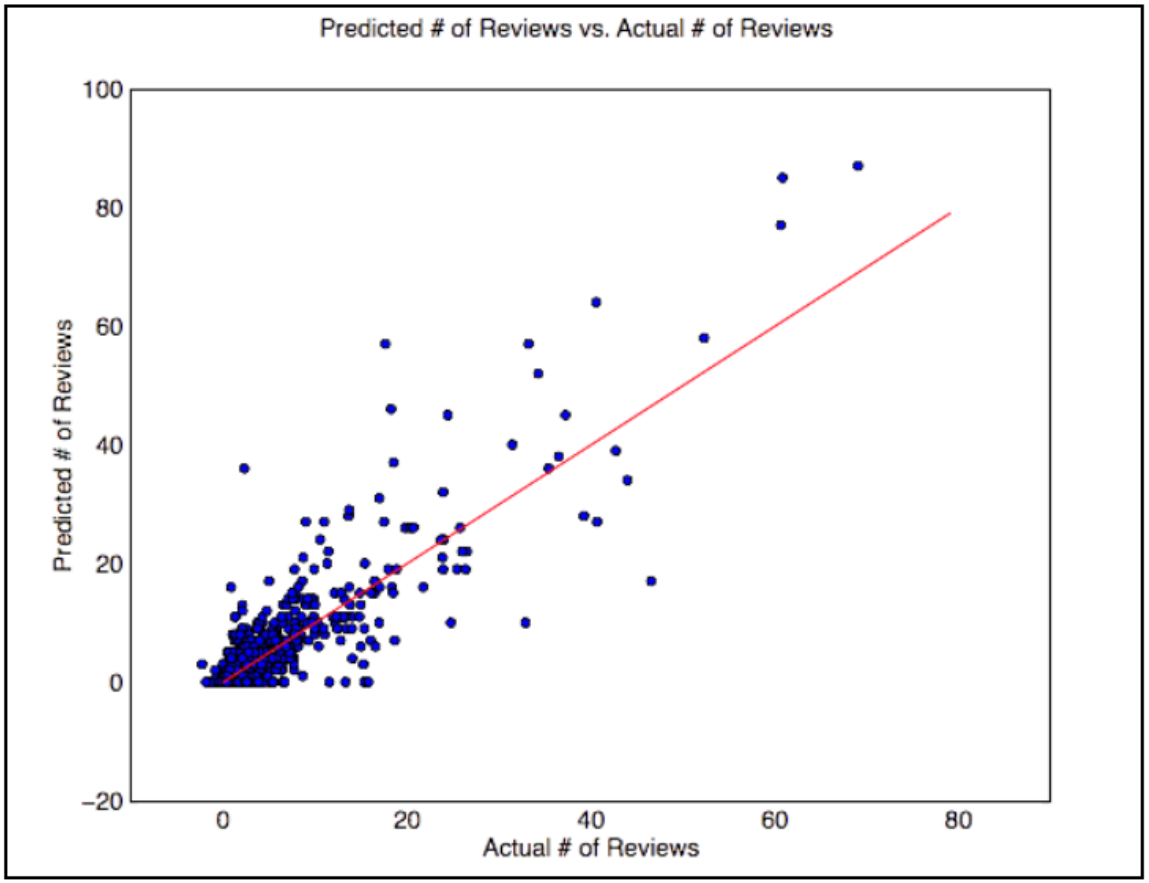


- Clusters were interpreted by their center means into labels that humans can understand.
  - basic, happy, long timers, angry, early adopters and power users.

*Source: http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_InferringFuture.pdf

Prediction

- Features are selected and scaled.
- Support Vector Regression (SVR) is used as regression model.
- **Prediction Task:** Given metadata about a business and all reviews in the Yelp database logged before a target date, predict the number of reviews that will be received for that business during the 6 month time period starting from the target date.



*Source: http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_InferringFuture.pdf

Improving Restaurants by Extracting Subtopics from Yelp Reviews*

- **Motivation:** To point out demand of customers from a large amount of reviews, with high dimensionality.
  - how can a restaurant point out the demands of its customers from a large amount of reviews?

- **Idea:** discovers latent subtopics from Yelp restaurant reviews by running an online Latent Dirichlet Allocation (LDA) algorithm.

  - These topics can provide meaningful insights to restaurants about what customers care about in order to increase their Yelp ratings, which directly affects their revenue.

  - By breaking these reviews down into latent subtopics using LDA, predicting a restaurant's star rating per hidden topic is possible.

  - These ratings per hidden topic allow us to pinpoint the reasons for a restaurant's Yelp rating, other than food quality.

*** iConference 2014 Berlin. Source:** http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_InferringFuture.pdf

Word Distribution of Topics

Lunch	Healthiness	American 1	Decor
8.0% lunch	7.0% menu	7.6% potatoes	2.9% patio
7.5% salad	4.4% options	5.5% rib	2.8% inside
6.6% sandwich	2.9% fresh	4.7% mashed	2.7% seating
4.0% chicken	2.7% vegetarian	3.7% prime	2.2% table
Service	**Location**	**American 2**	**Value**
4.3% service	7.9% phoenix	9.5% fish	7.3% portion
3.5% food	3.1% miss	4.9% chips	5.3% price
3.0% asked	2.7% area	4.3% sliders	3.2% small
2.9% server	1.9% town	3.6% son	1.9% huge

Service, for example, is made up of words such as ”service,” ”asked,” and ”server,” with corresponding numbers 4.3, 3.0, and 2.9 which represent the percent that each word makes up of that subtopic.

Breakdown of Topics Over All Reviews

Of the 50 subtopics generated from the Online LDA algorithm, some of the more interesting and more frequently occurring topics are listed below.

service	8.8%	wait	1.64%
value	5.85%	music	0.77%
take out	3.64%	breakfast	0.59%
décor.	2.99%	dinner	0.50%
healthiness	2.62%	lunch	0.50%

- Users care the most about service of all of these subtopics, making up 8.8% of all reviews.
- Users also care greatly about value, take out and decor.
- Temporal topics also arise, such as the breakfast, lunch, and dinner categories.

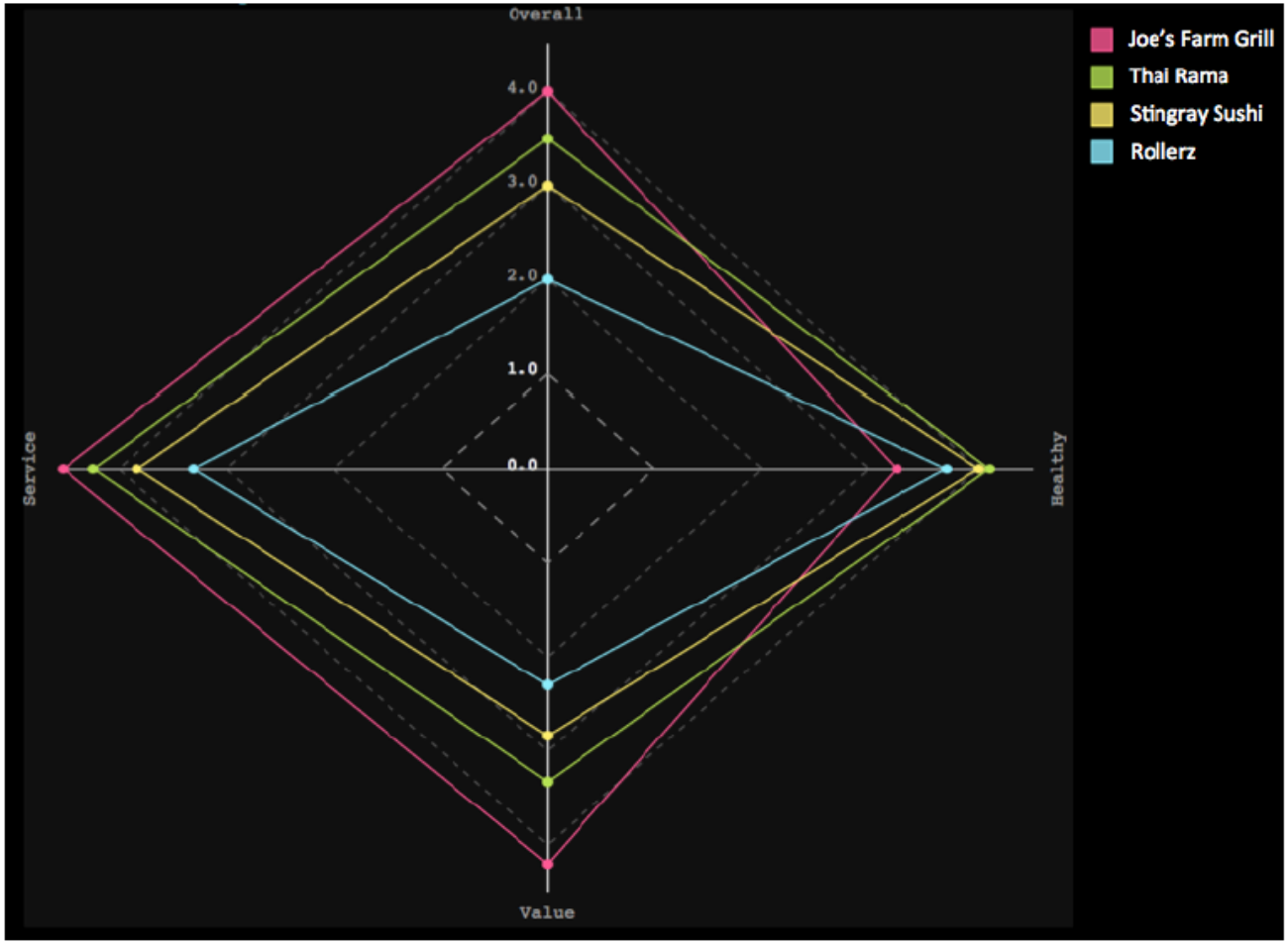
Predicting Hidden Topic Stars of four restaurants

Consider all reviews for a restaurant that contained the given topic and averages over all of these review ratings to get the hidden topic rating.

Joe's Farm Grill:
Overall rating : 4.0
Service rating: 4.513
Value rating: 4.203
Healthiness rating: 3.25.

- The lower predicted rating from the healthiness subtopic is pulling the overall rating for Joe's Farm Grill down.

- Recommend that Joe's Farm Grill change some of their healthiness choices in order to bring their Yelp rating up.



Clustered Layout Word Cloud for User Generated Review*

❑ **Motivation:** Reading large quantities of reviews is a difficult and time-consuming task. In this situation, a visualization that summarizes the user generated reviews is needed for perusing reviews.

❑ **Objective:** The paper presents the concept of ***clustered layout word cloud; a text visualization that would assist in making decision quicker based on user generated reviews.***

❑ **Why Word-Cloud?**

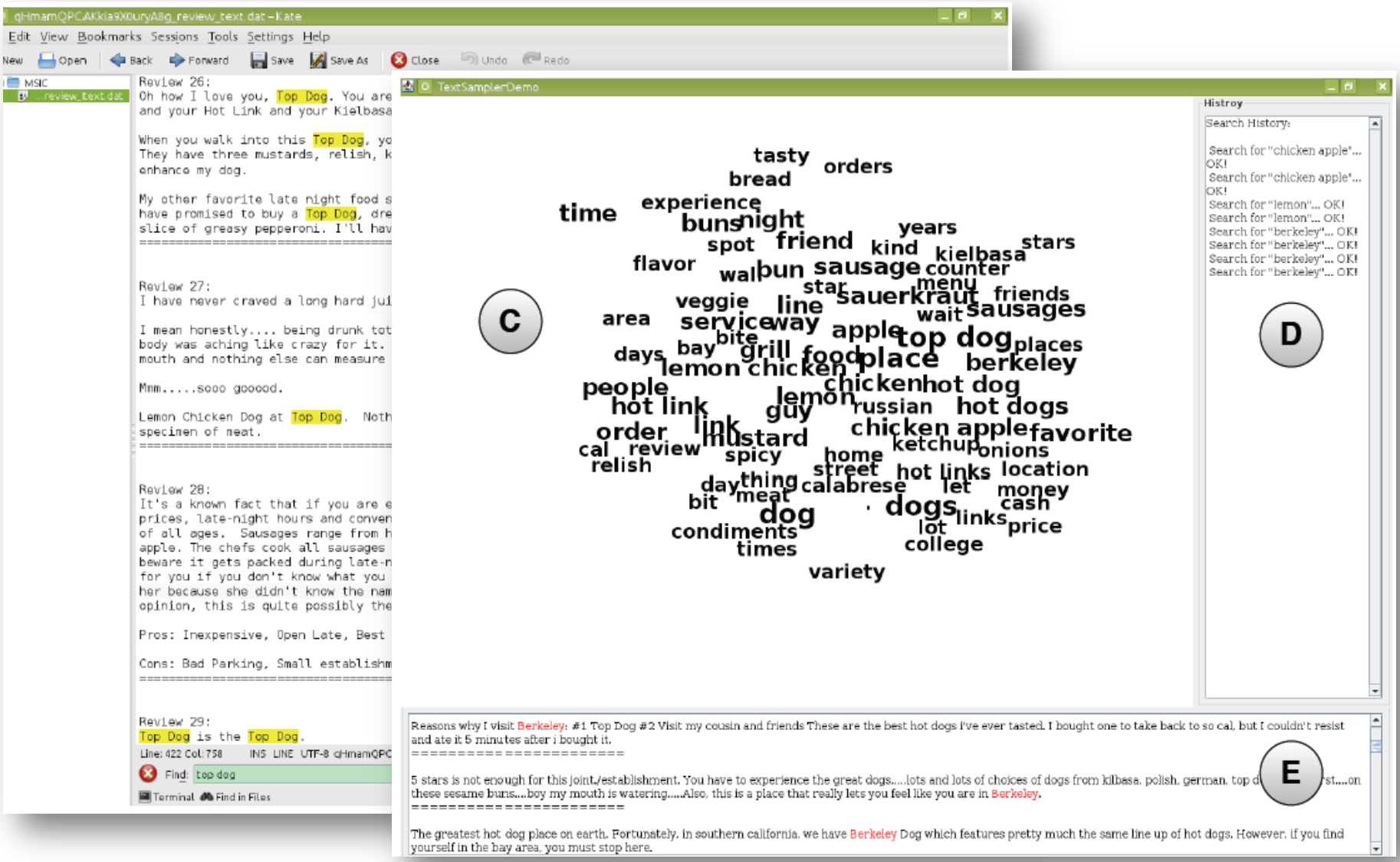
- Word clouds have the potential to process user generated reviews and provide more context than only quantitative information.

❑ **Why Clustered Word-Cloud, instead of random layout Word-cloud?**

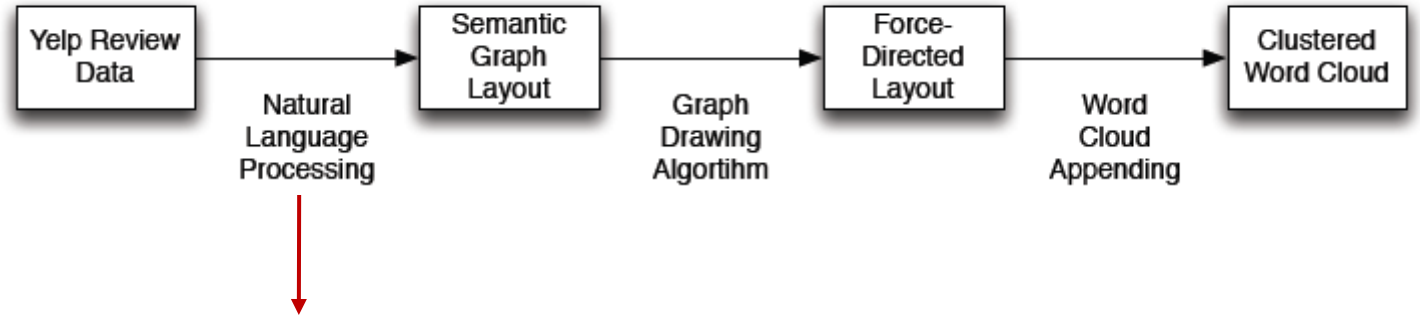
-The random layout of word clouds can require significant mental demands when trying to understand the review content and leads to a higher cognitive load.

* Graphics Interface 2014 Montreal

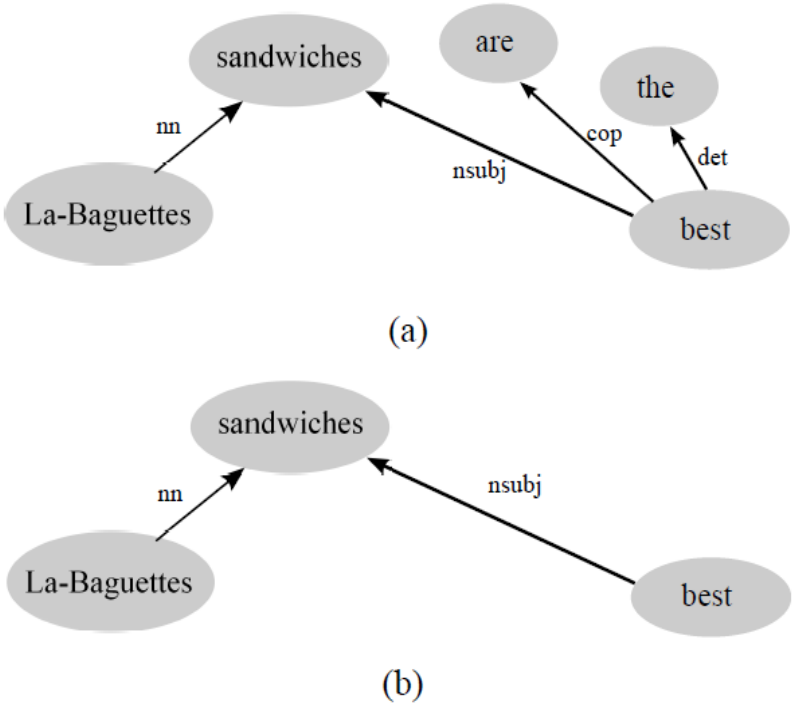
Does Visualization make any difference?



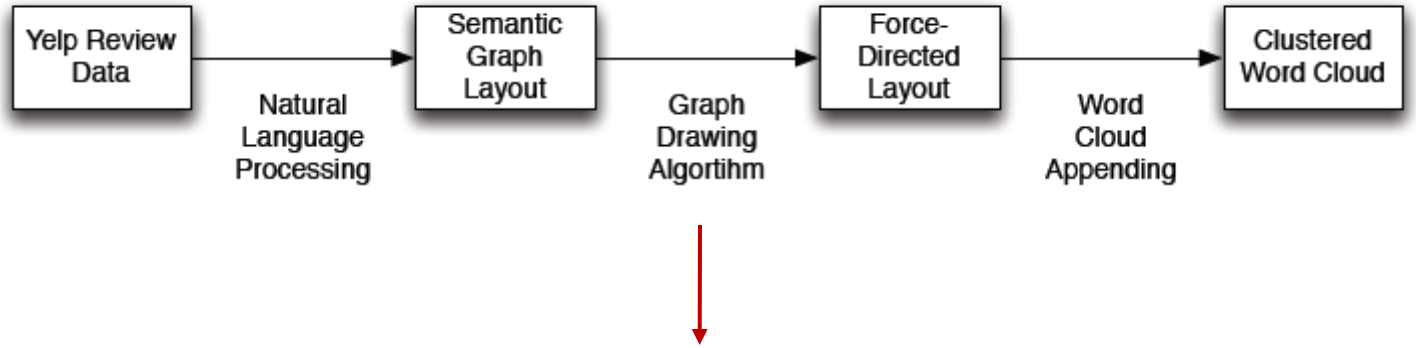
The Processing Pipeline for Clustered Layout Word Cloud



1. The review content for a specific restaurant is first extracted from the raw dataset and chunked into sentences.
2. Sentences are parsed based on grammatical relations and eventually the relationship information are filtered to form a context graph from the user generated review content for a specific business/restaurant.
3. From the sentence-level grammatical relation, a concatenation of the relations is performed to form a graph for all the reviews of the restaurant the user is querying.

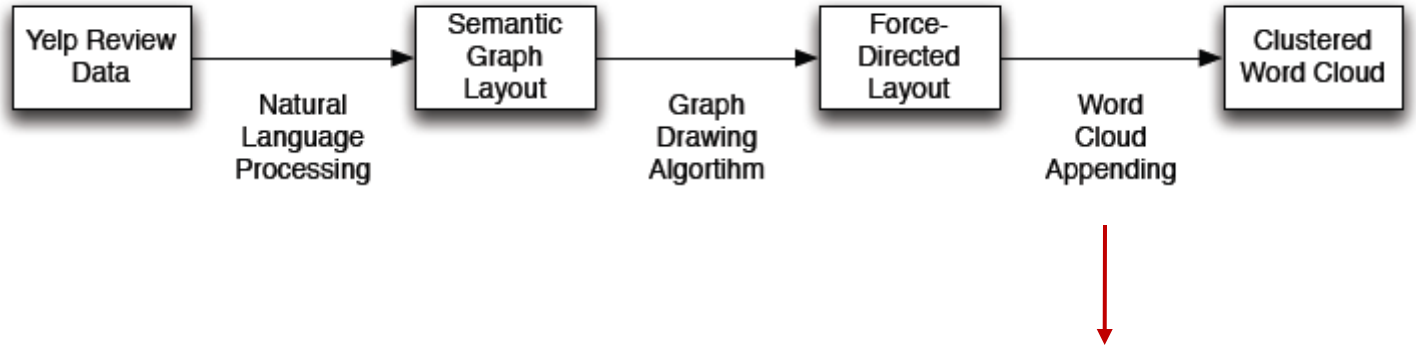


The Processing Pipeline for Clustered Layout Word Cloud



From the semantic graph layout, the **LinLogLayout energy model** layout algorithm is used to create a ***force-directed graph layout to provide a basis for the clustered layout word cloud.***

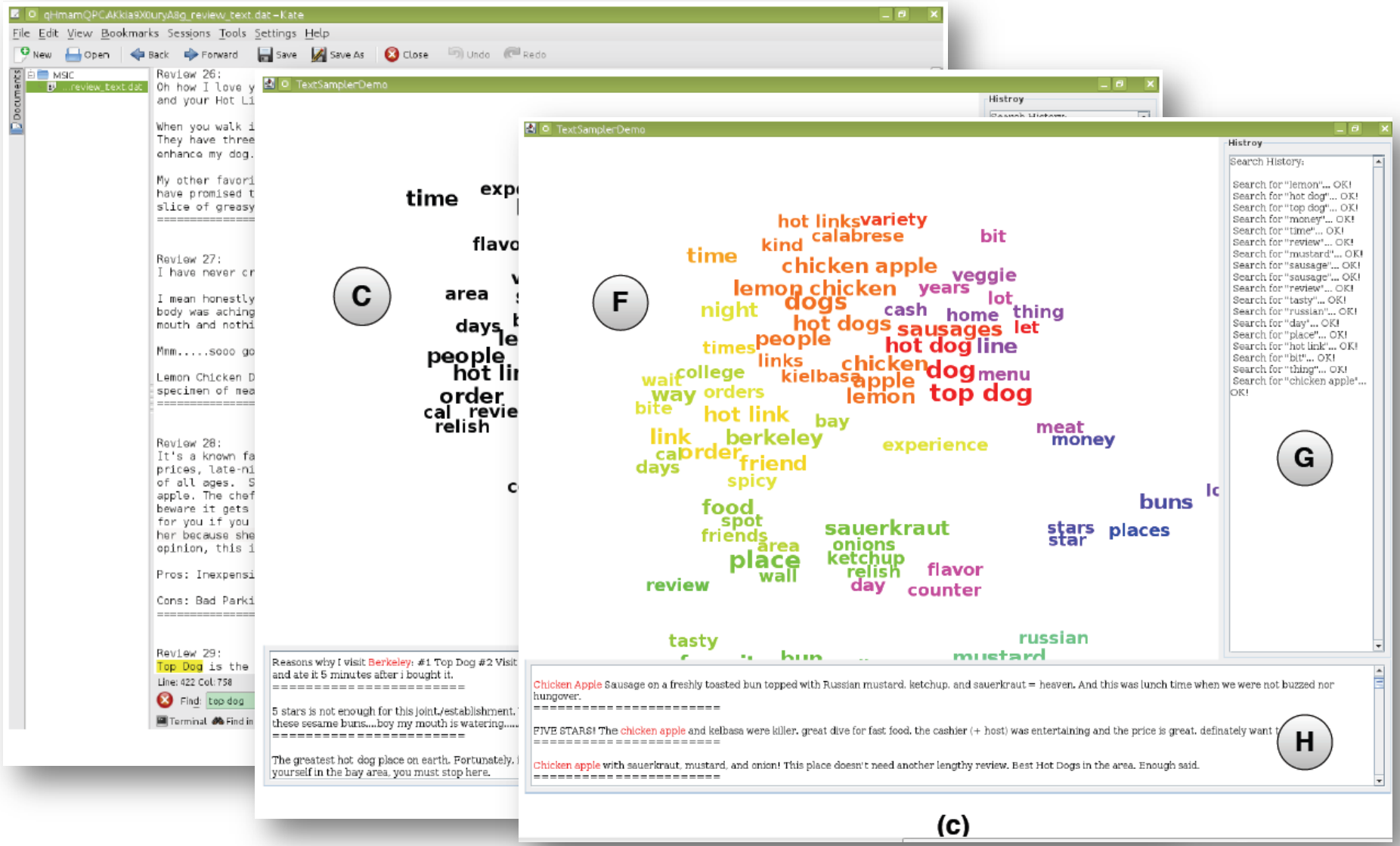
The Processing Pipeline for Clustered Layout Word Cloud



1. A word cloud generation approach is then applied to the force-directed graph layout in order to display clustering information.

2. Color encoding and clickable interaction were appended to the word cloud to provide faster recognition of clusters and achieve the ability to conduct quick keyword search respectively.

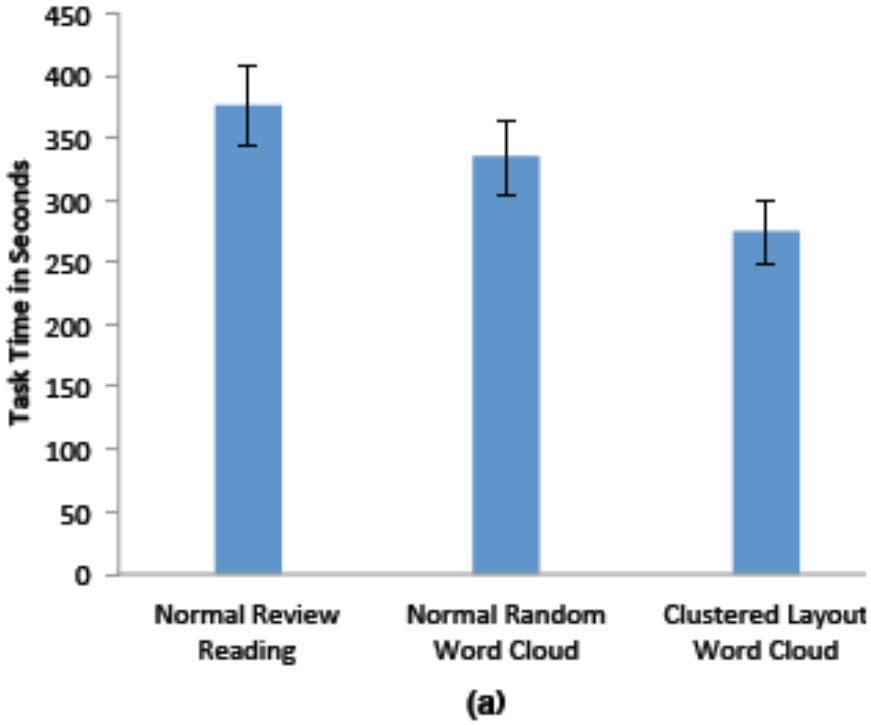
Evaluating the Significance of Clustered Layout Word Cloud



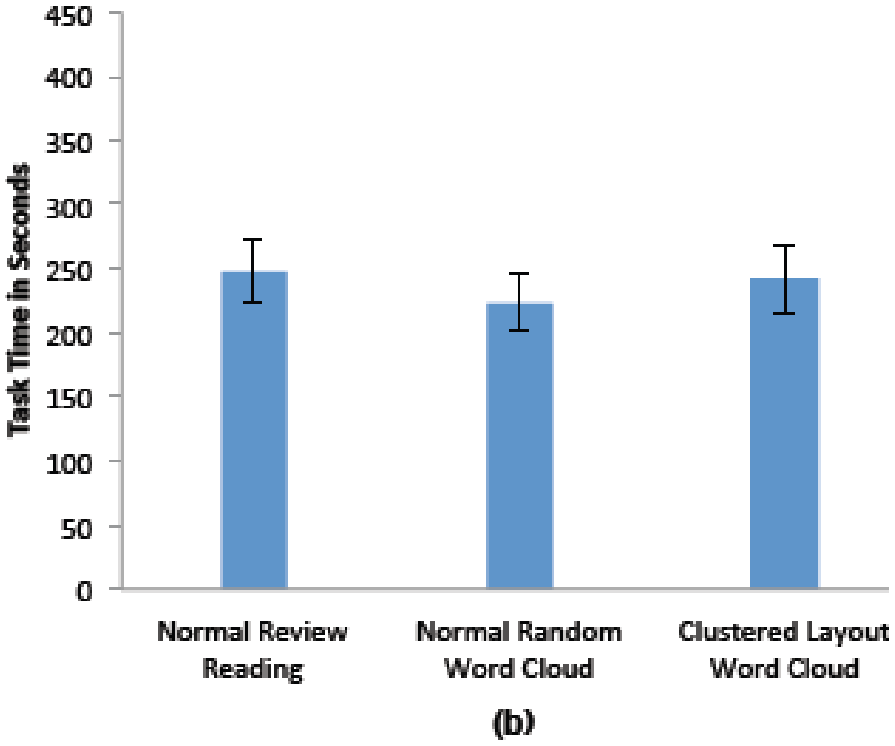
Task Design and Study Procedure

Task Type	Approach	Data	Task
Decision Making	Normal Review Reading	Good Good Pair 1	Which restaurant will you go?
	Normal Review Reading	Good Bad Pair 1	
	Normal Random Word Cloud	Good Good Pair 2	
	Normal Random Word Cloud	Good Bad Pair 2	
	Clustered Layout Word Cloud	Good Good Pair 3	
	Clustered Layout Word Cloud	Good Bad Pair 3	
Feature Finding	Normal Review Reading	Restaurant 1	Food Feature
	Normal Review Reading	Restaurant 2	Non-food Feature
	Normal Random Word Cloud	Restaurant 3	Food Feature
	Normal Random Word Cloud	Restaurant 4	Non-food Feature
	Clustered Layout Word Cloud	Restaurant 5	Food Feature
	Clustered Layout Word Cloud	Restaurant 6	Non-food Feature

Comparing Decision Making Task Completion Time

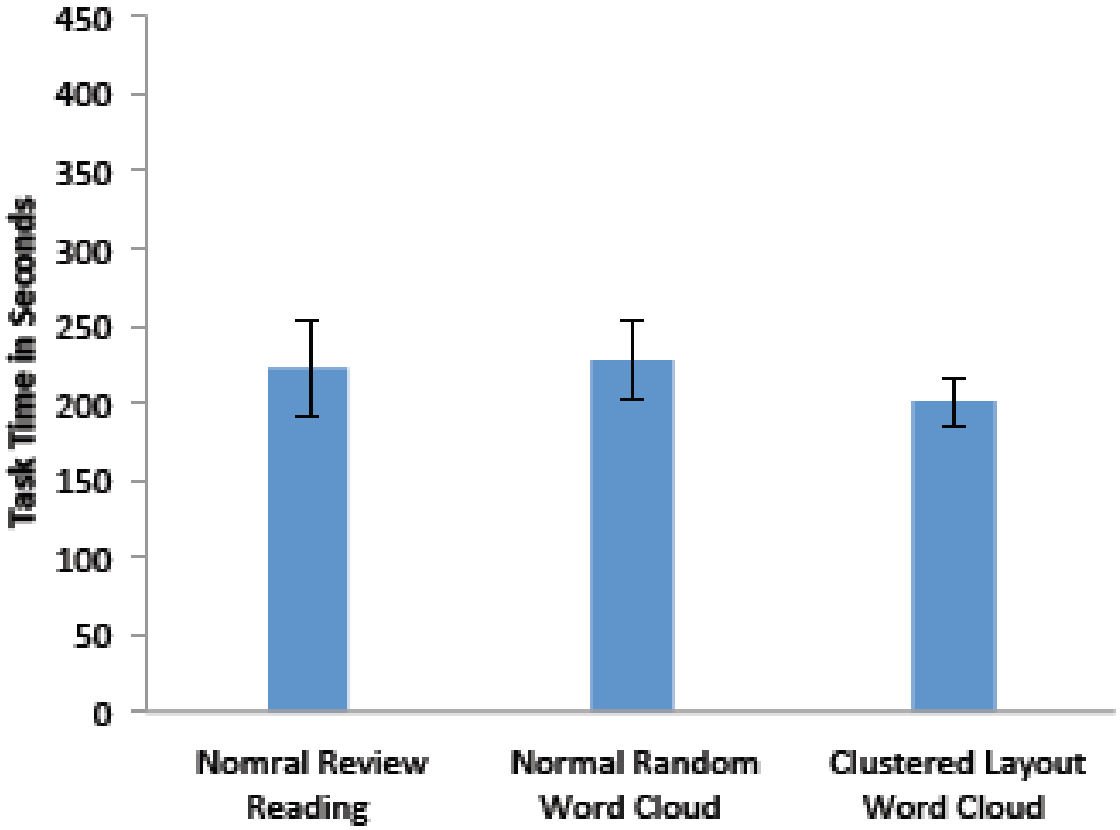


Good-good pair



Good-bad pair

Comparing Feature Finding Task Completion Time



Further Reading ...

- ✓ **Valence Constrains the Information Density of Messages**
  David W. Vinson, Rick Dale. University of California, Merced.

  Source: http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_InformationDensity.pdf

- ✓ **On the Efficiency of Social Recommender Networks**
  Felix W. Princeton University.

  Source: http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_NetworkEfficiency.pdf

Further Reading ...

- ✓ **Personalizing Yelp Star Ratings: a Semantic Topic Modeling Approach**

  Jack Linshi. Yale University.

  Source: http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_PersonalizingRatings.pdf

- ✓ **Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text***

  Julian McAuley and Jure Leskovec. Stanford University.

  Published in ACM RecSys '13 Proceedings

ActMiner* and LANet*:

A Case Study of Location-aware Review Analytics

* Sahisnu Mazumder, Dhaval Patel and Sameep Mehta: **ActMiner*: Discovering Location-specific Activities from Community-Authored Reviews***. In **DaWaK 2014**.

* Sahisnu Mazumder: **LANet*: An Enriched Knowledgebase for Location-aware Activity Recommendation System.*** **[Final Year M. Tech. Thesis, 2014.]**

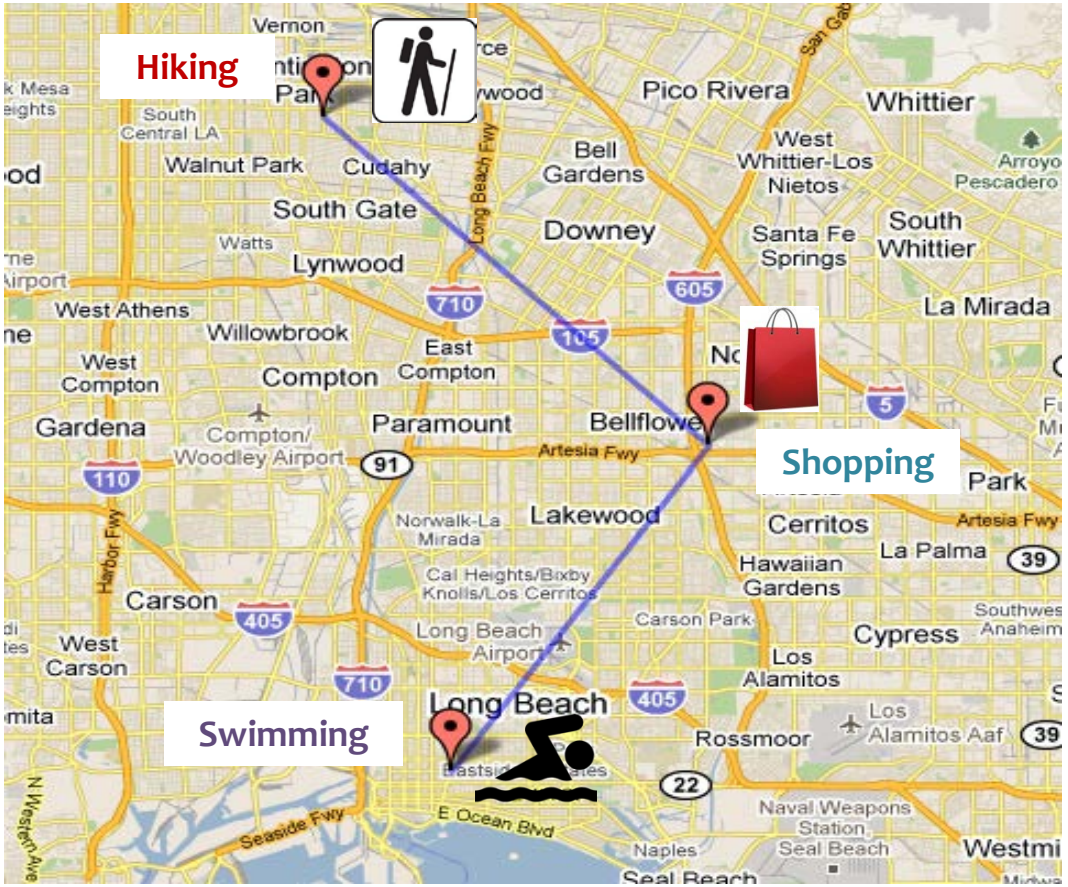
Why we Travel?

"We wander for distraction, but we travel for fulfillment."

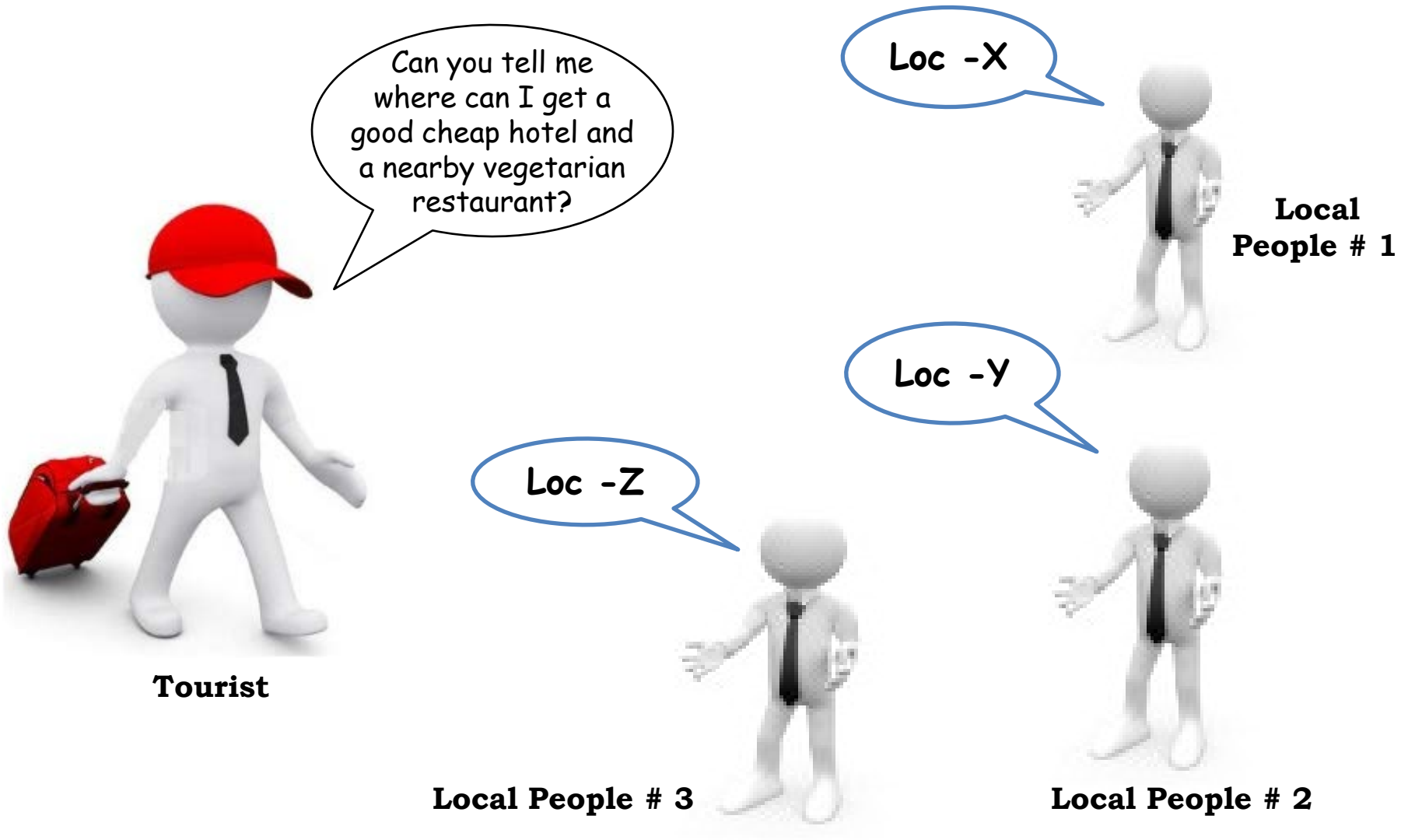
-Hilaire Belloc.

Whenever we visit some place, always there is some purpose associated with it……that is

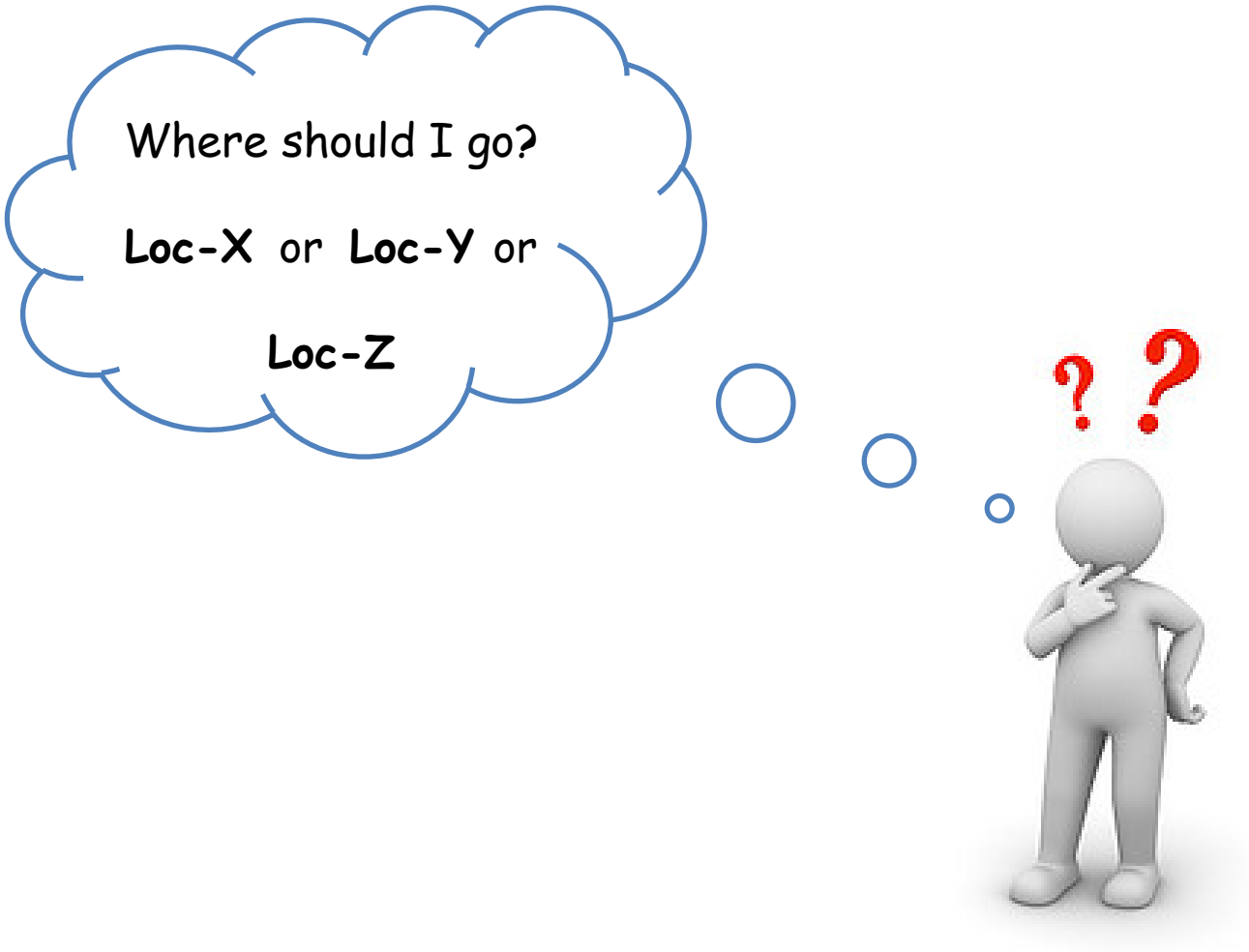
To Perform some Activity of our Interest



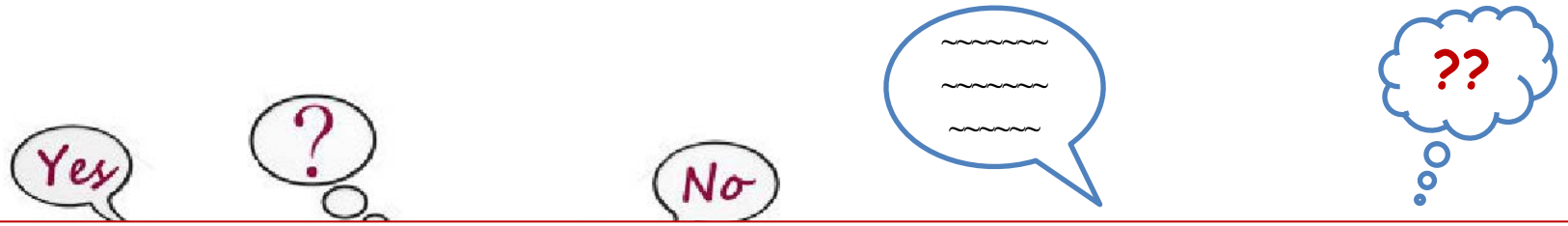
A Situation we often face while Travelling…



And the Next Scene!



Moreover…

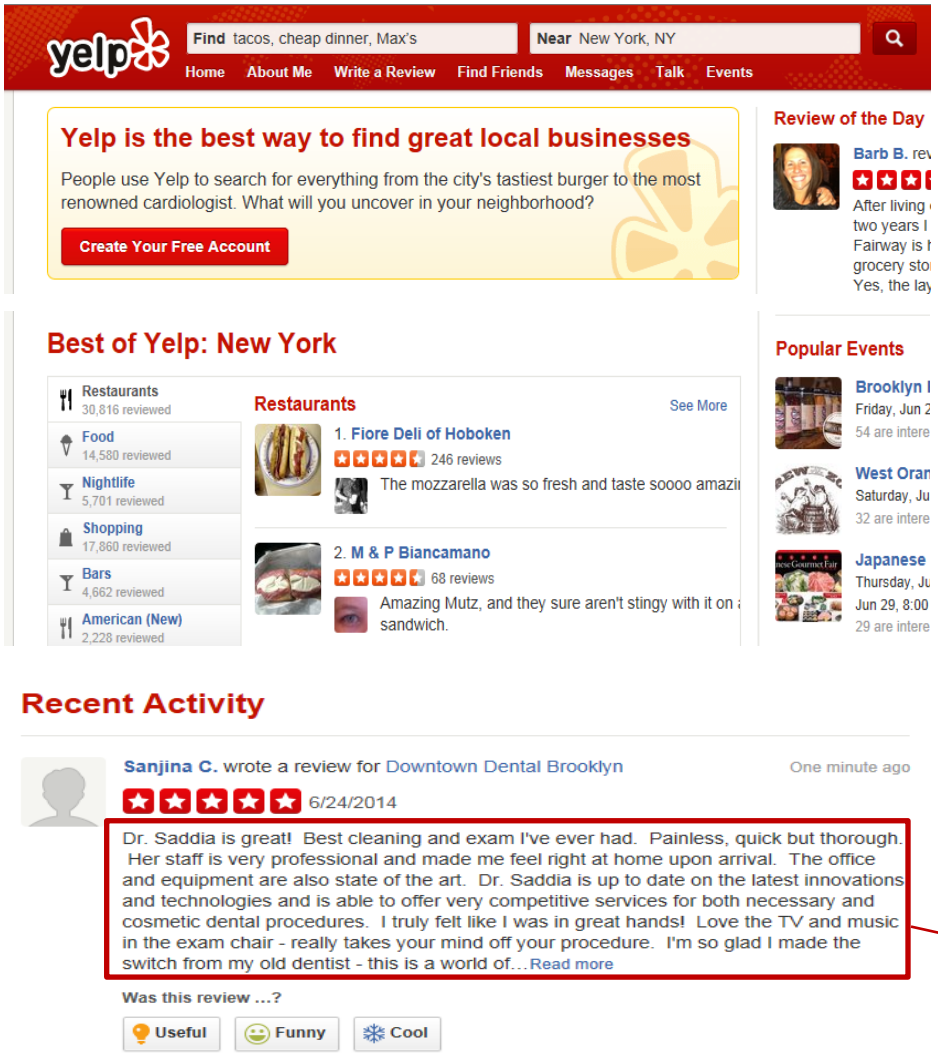


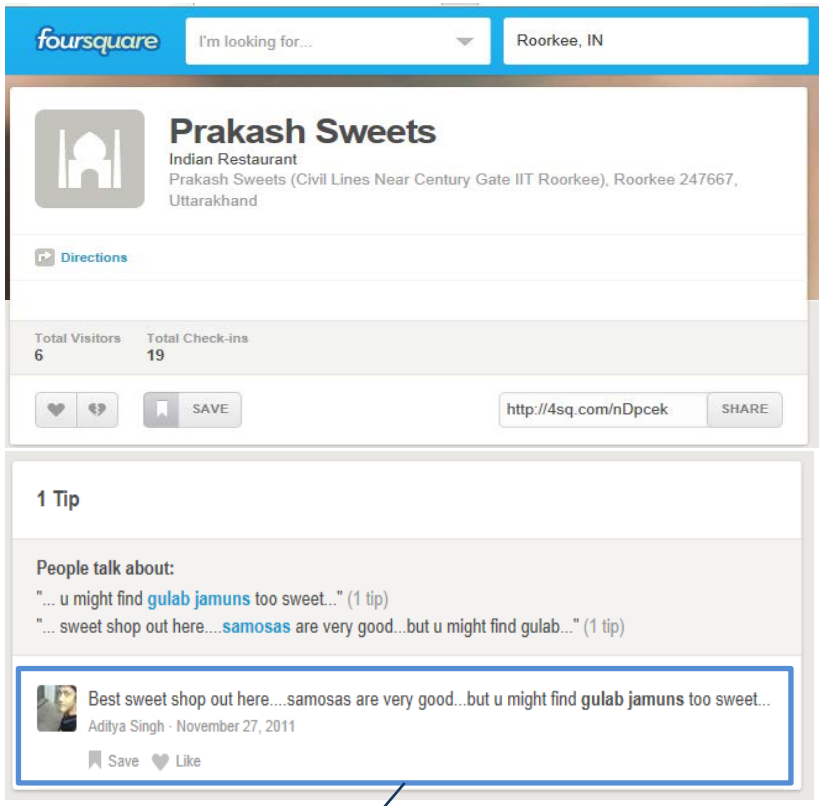
So, we need a system that can recommend us the best nearby locations for performing our intended activities [defined as (verb, noun/noun phrase) pair].

And for that we need to discover **the knowledge of location-specific activities.** **But, How?**

- Often the suggestions given by Local people are **diverse** and **biased.**
- **Conversational Language** is also an issue.

LBSNs can help Us in Activity Discovery!





Location-aware Reviews
– A resource for discovering Location-specific activities.

But, is there any Existing Solution?

Baseline Approach* for activity Discovery

verb | Nouns | Noun Phrase

“

(h

Limitations :

- Doesn't discover the **complete Activity set.**

So, There exists a couple of serious **limitations of the existing approach**

and

We need to design an effective solution to overcome these limitations.

(watch, cricket match) activity is not relevant to ... restaurants”.

In the sentence ... food has been served by the waiter

(make, food) and (prepare, food) activities are redundant to each-other.

Baseline: (served, waiter) **Actual: (served, food)**

*Dearman et. al., *Identifying the activities supported by locations with community-authored content*, ACM UbiComp 2010.

The Problem We have solved ....

Input: Location-aware Reviews and Category of Locations

Output: List of Location-specific Activities

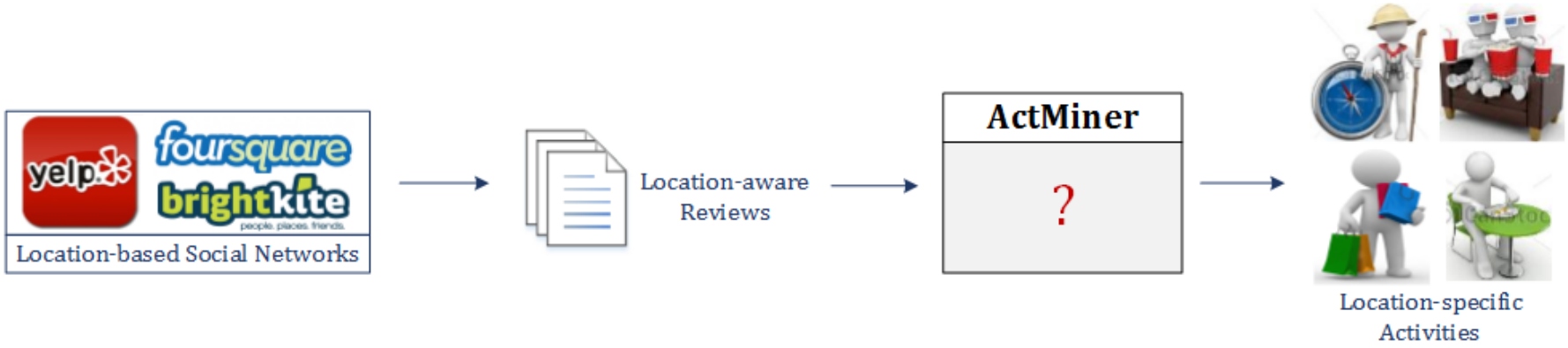
cat_i	{"Restaurant","Hotel" }
Review	**Text**
R_i^1	"Car driving and sometimes, trying out good new foods are my hobbies. Yesterday I came here for dinner. The food was served late and I had to wait for a long time. But, I enjoyed chicken tikka masala and had a great time there."
R_i^2	"Had dinner with my old friends. The food was awesome... I liked butter nun very much."
R_i^3	"Yesterday, we celebrated my aunt's birthday in the banquet hall. Everyone enjoyed a lot..."

Activity	Frequency
(try, food)	1
(have/come, dinner)	2
(serve, food)	1
(wait, time)	1
(enjoyed, chicken tikka masala)	1
(like, butter nun)	1
(celebrate, birthday)	1

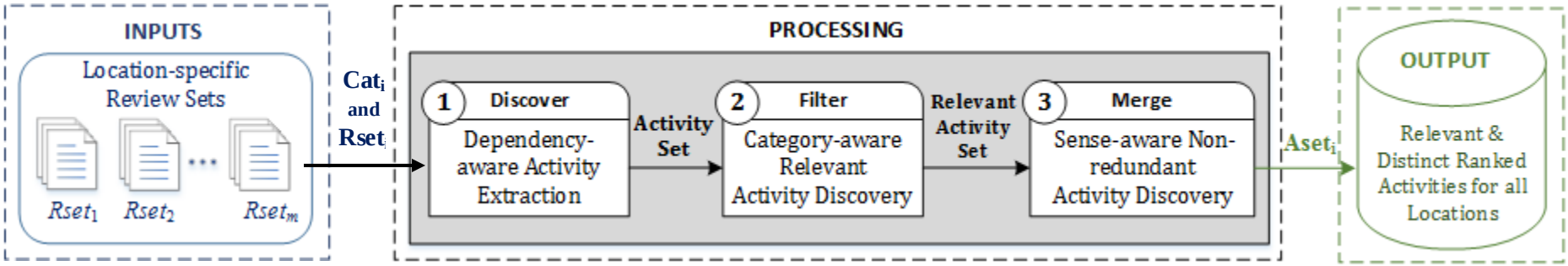
Problem Statement

Given the **categories** and **review sets for all *m* locations,** *our propose Solution discovers the set of location-specific* ***meaningful, relevant*** *and* ***non-redundant activities*** *for each location.*

ActMiner : How it Works?



Overview of ActMiner

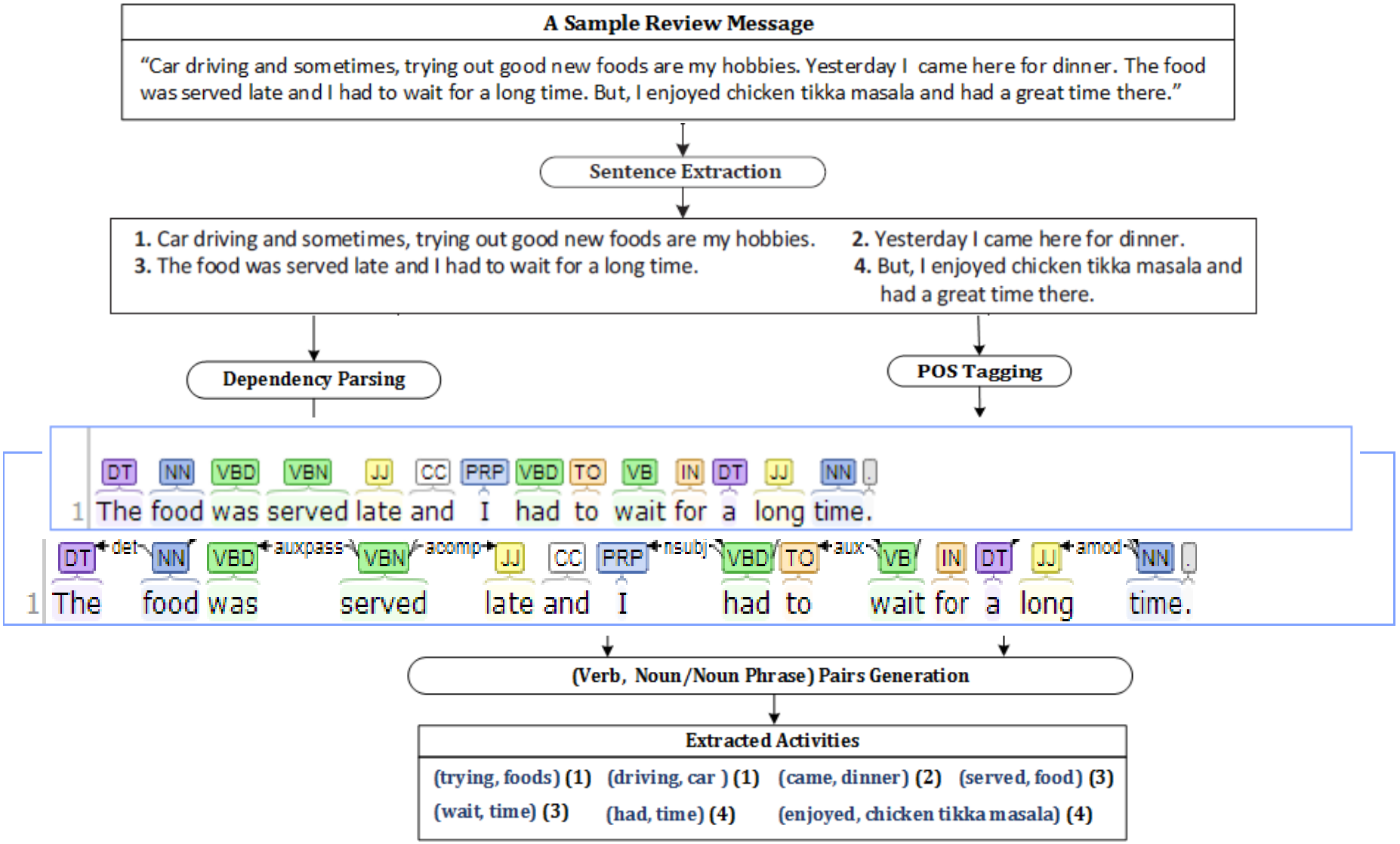


1. Discovers potential (verb, Noun/ Noun Phrase) Pairs that represent meaningful activities.

2. Discovers relevant activities from output of phase-1 using ConceptNet and Category of the location.

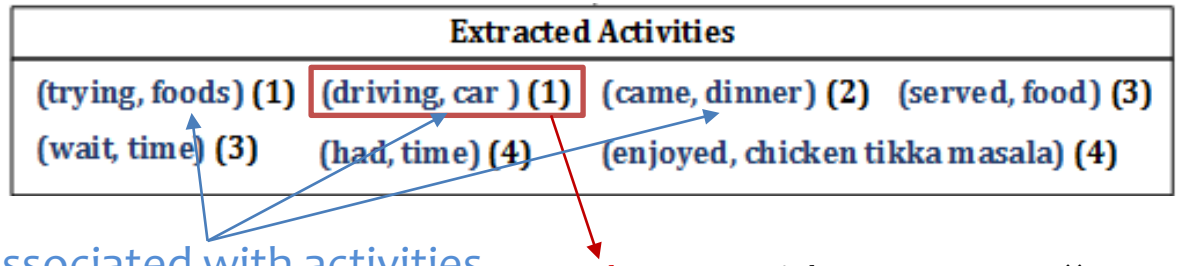
3. Discover redundant activities using sense of an activity and merges them into single one.

Dependency-aware Activity Extraction



Do we need further processing?

We have discovered meaningful activities, but are the inferred activities all relevant ?



So, we need a effective mechanism to discover the relevant activities from the Inferred ones... But How ?

Clue: Every activity is associated with a Concept and if the Concept is related to the Category of the location, then the activity is relevant.

Solution: **ConceptNet** and a ***novel*** technique:

Category-aware Concept Hierarchy (CCH)

Power of ConceptNet

food

sushi – *IsA* → food
Kinds of food : sushi

A Problem!

- ✓ Often **direct relation between two concepts** is not available.
- ✓ As of April 2012, ConceptNet contains-

  12.5 million edges which connects 3.9 million concepts.

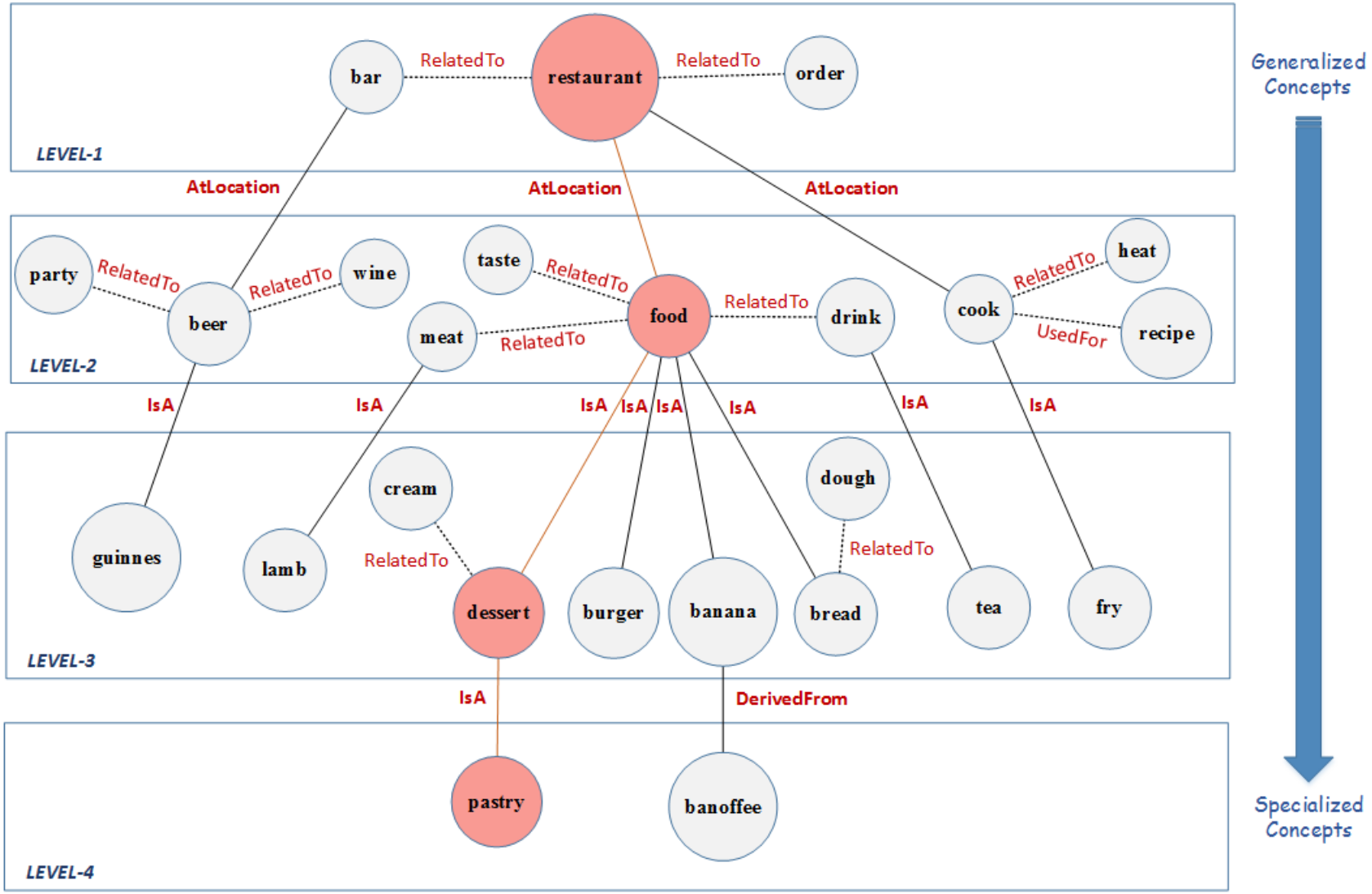
So, **Exhaustive Searching** of ConceptNet is **Computationally Expansive**!

…… **Motivation** for CCH Construction

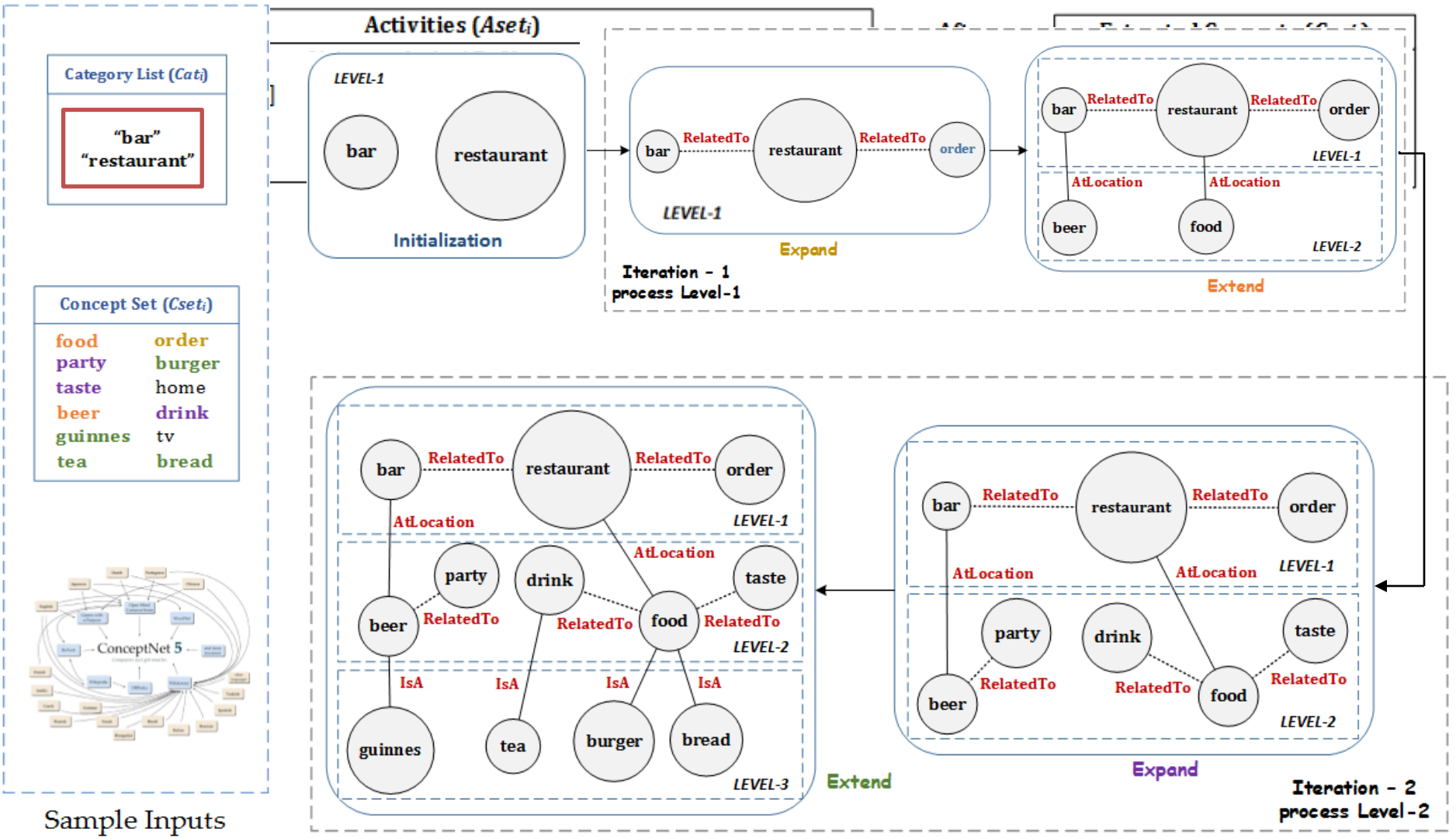
food – *AtLocation* → restaurant
Somewhere food can be is a restaurant.

pizza – *IsA* → food
pizza is a kind of food.

Category-aware Concept Hierarchy (CCH)



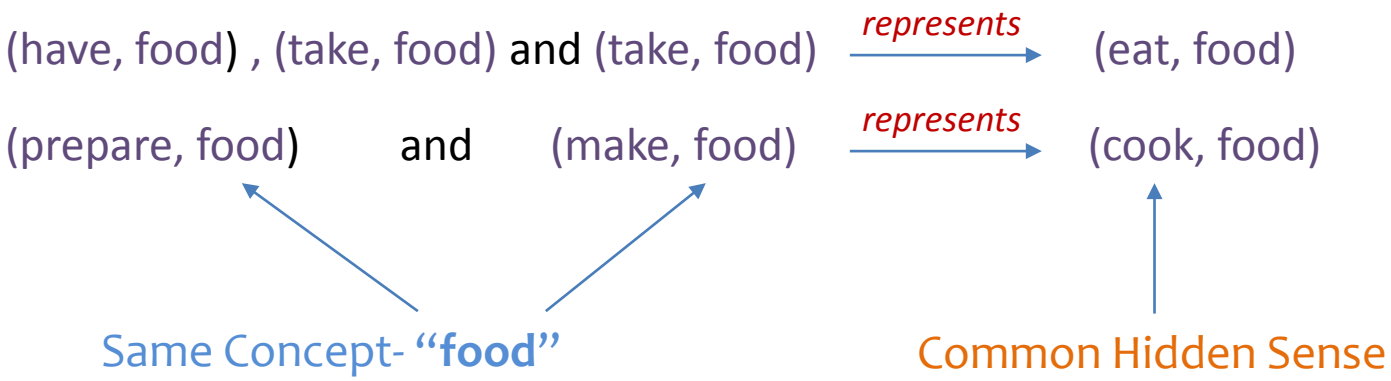
Formation of CCH



Are we done ?

Now, we have discovered meaningful as well as relevant activities.

But, there may be redundant activities as well.



So, these activities are redundant and should be merged …

But, How can we Detect the **Common Hidden Sense** of a set of activities???

Sense-aware Non-redundant Activity Discovery-I

ConceptNet can help us in **redundant activity discovery** and in their **merging Process!**

Common Hidden Sense Discovery

have food — *UsedFor* → eat
having food is for eating

eat — *RelatedTo* → take food
eat is related to taking food

eat — *RelatedTo* → get food
eat is related to get food

But, How the merging will be done ?

Sense-aware Non-redundant Activity Discovery-II

The Answer is **"Sense-base Activity Clustering".**

But, ConceptNet can't solve the problem Alone !

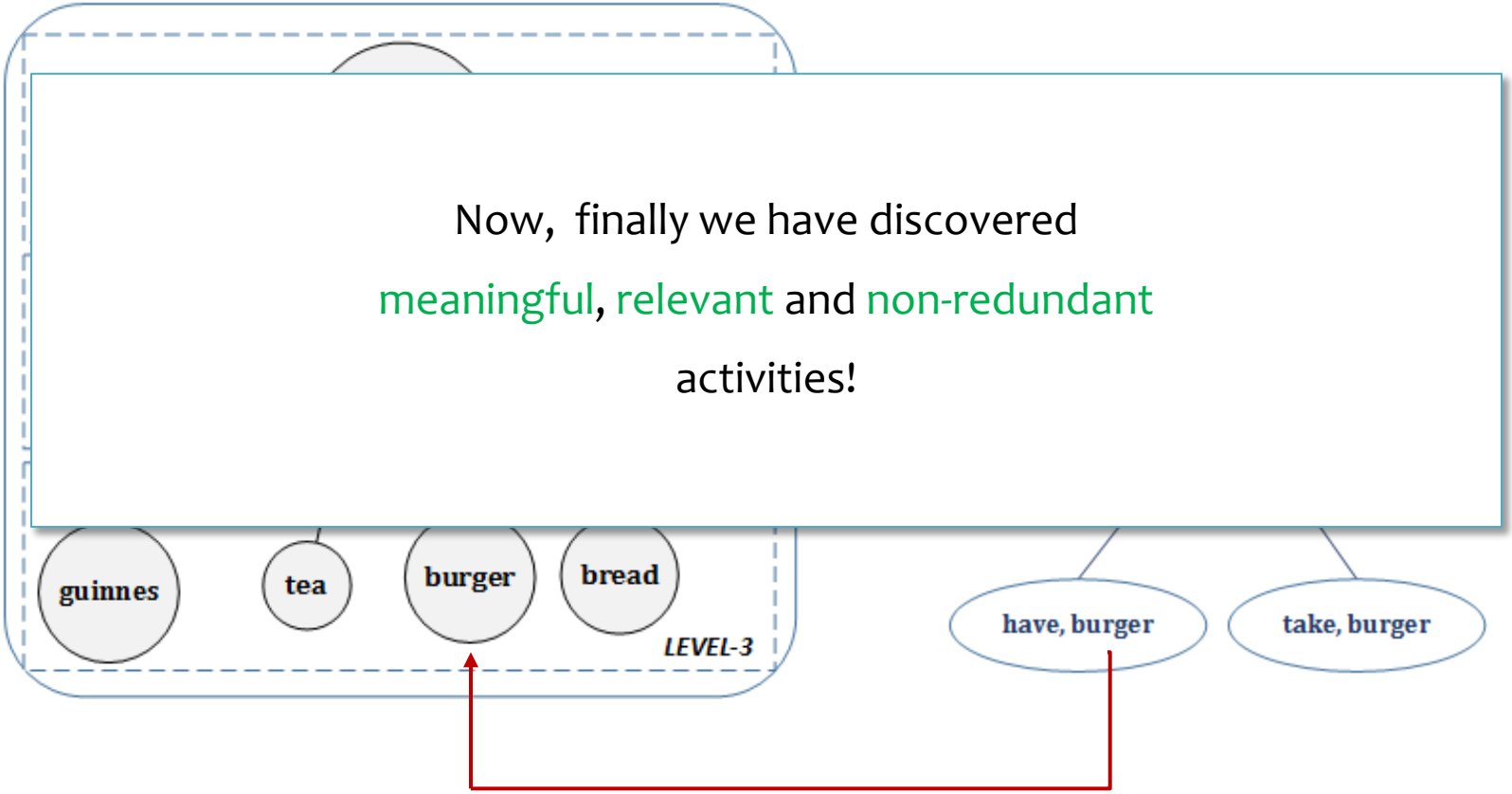
ConceptNet can only help us to merge **redundant activities** which are **associated with generalized concepts…not in specialized activity merging.**

Here, CCH plays the GAME!

Activity Cluster (take/have/get, food) Activity Cluster (make/prepare, food)

Reusing CCH in Activity merging ....

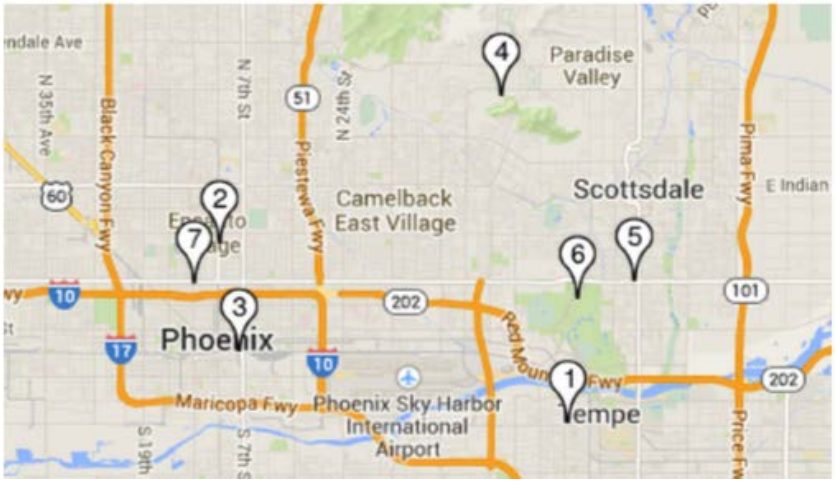
Consider two redundant activities **(have, burger)** and **(take, burger)**



How well ActMiner Performs?

Data Sets used in Evaluation

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



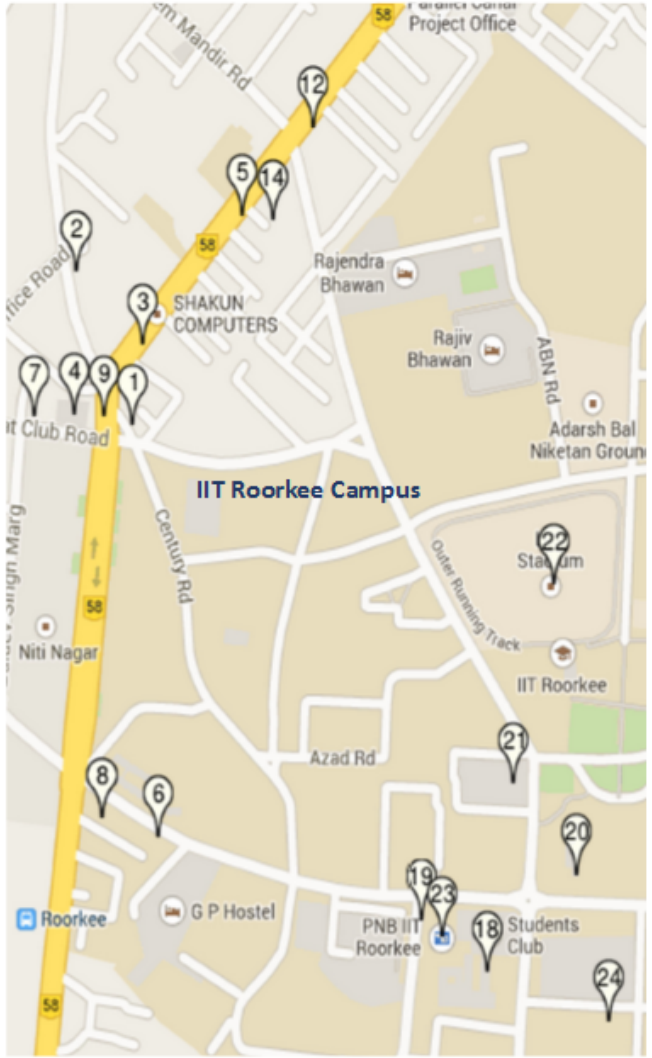
Yelp Data set collected from Yelp Dataset Challenge

A Generous sample of yelp data from the greater Phoenix, AZ metropolitan area with:

- **11,537 business locations.**
- **8,182 check-in sets.**
- **2,29,907 reviews.**
- **43, 873 users.**

Data Sets ...(Roorkee Dataset)

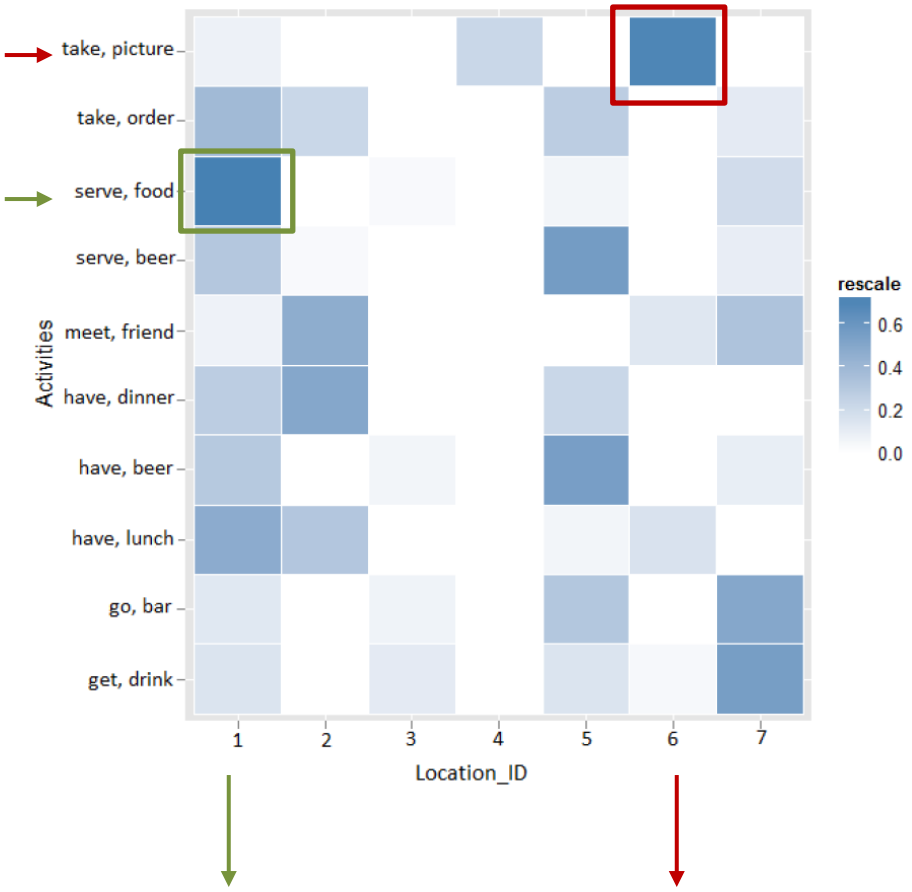
Loc_ID	Location Name	Categories	No. of Reviews
1	Prakash Sweets.	"Dessert Shop","Snacks"	61
2	Kundan Sweets.	"Dessert Shop","Snacks"	28
3	Prakash Hotel.	"Hotel", "Restaurant"	55
4	Hotel Royal Palace.	"Hotel", "Restaurant","bar"	38
5	Dominos Roorkee.	"Pizzaries"	46
6	Sizzlers.	"Restaurant"	13
7	Food Point.	"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



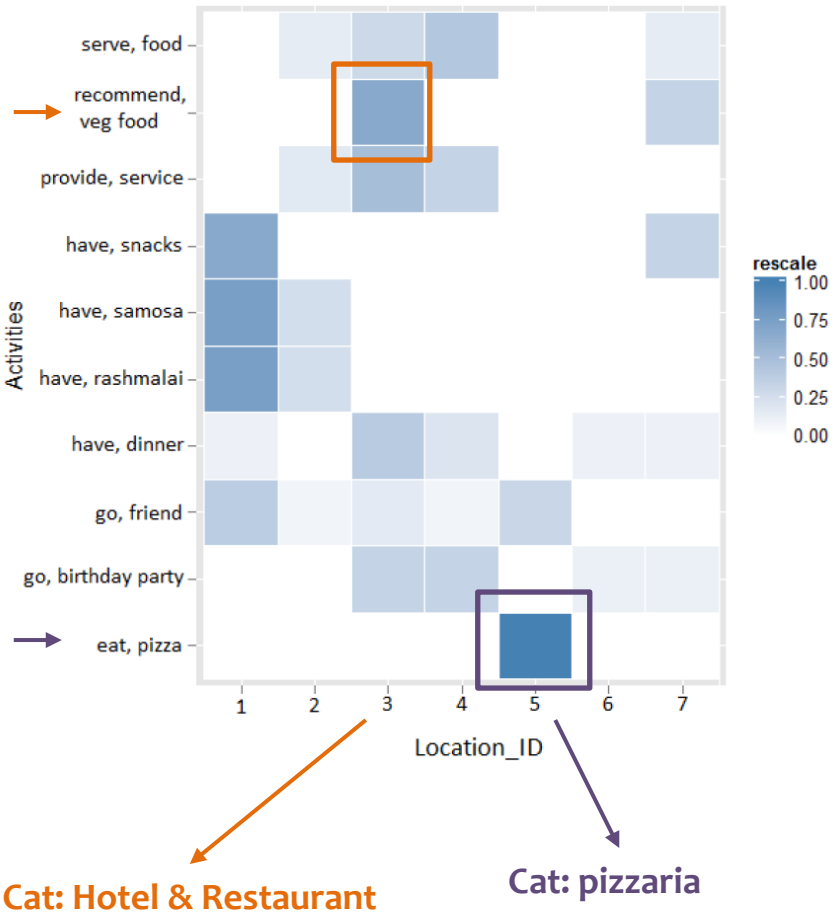
** These locations are not shown in the map due to space limitations*

Evaluating Correctness of ActMiner…

Activity Popularity on Yelp



Activity Popularity on Roorkee



Evaluating Accuracy of ActMiner

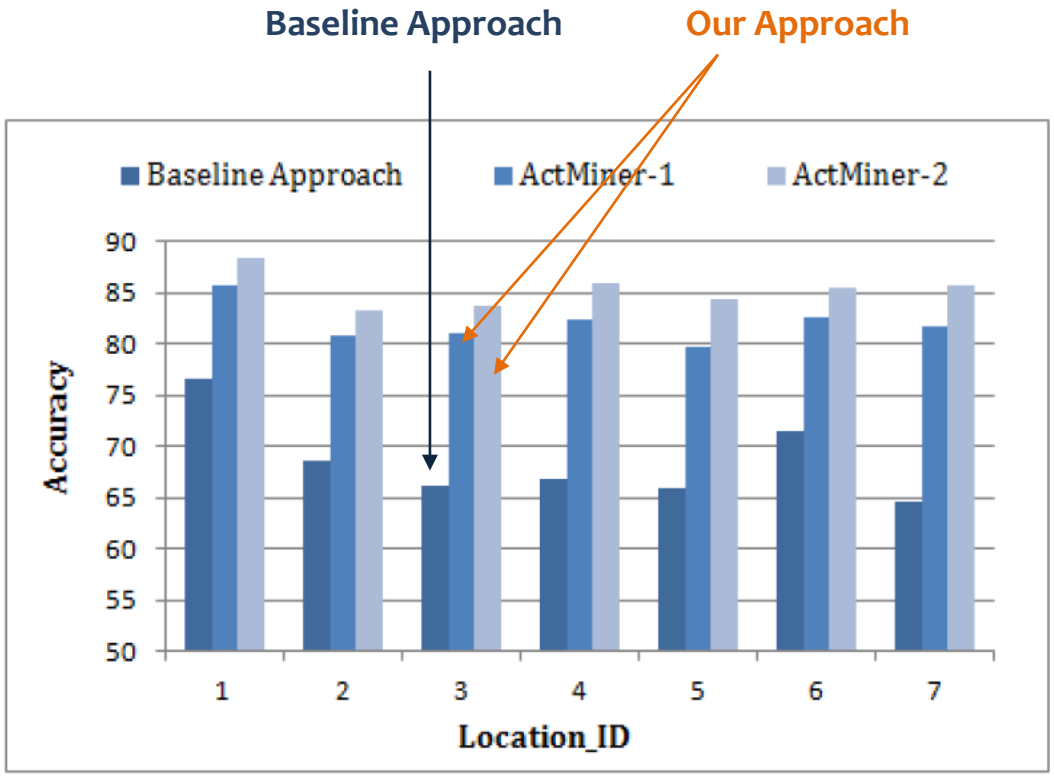
On Yelp Dataset

Avg. Accuracy Statistics

Baseline : 68.6%

ActMiner-1 : 82%

ActMiner-2: 85.23%



ActMiner-1 = Dependency-aware Activity Extraction

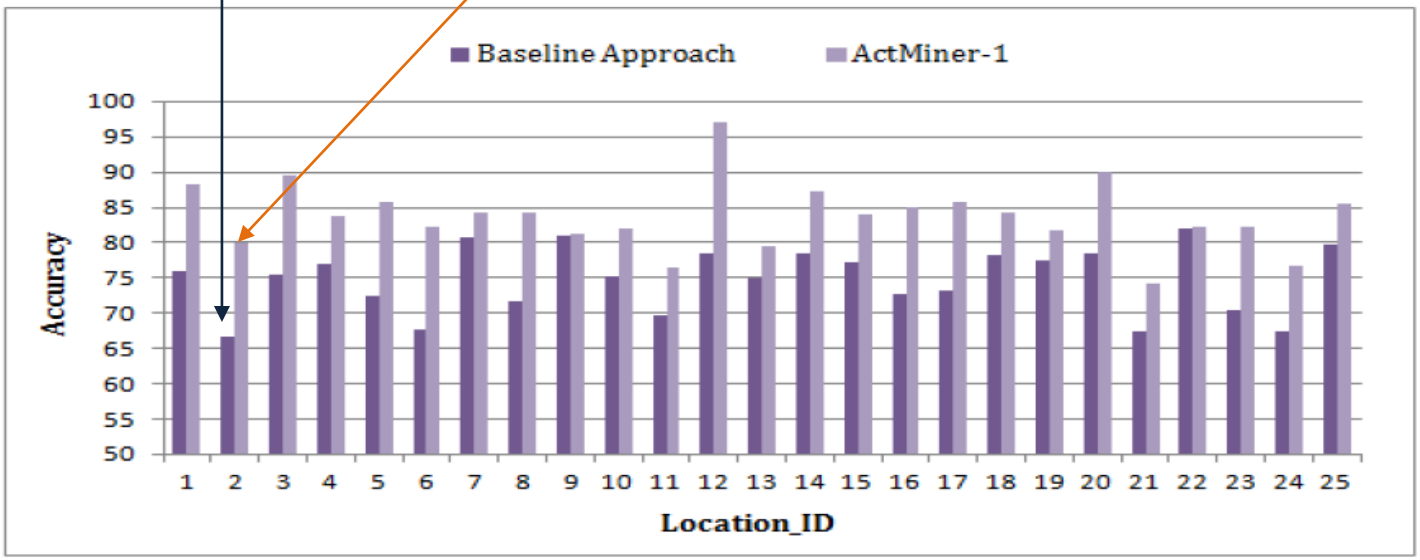
ActMiner-2 = Dependency-aware Activity Extraction + Category-aware Relevant Activity Discovery

Evaluating Accuracy of ActMiner

On Roorkee Dataset

Baseline Approach

Our Approach



ActMiner-1 = Dependency-aware Activity Extraction

Avg. Accuracy Statistics

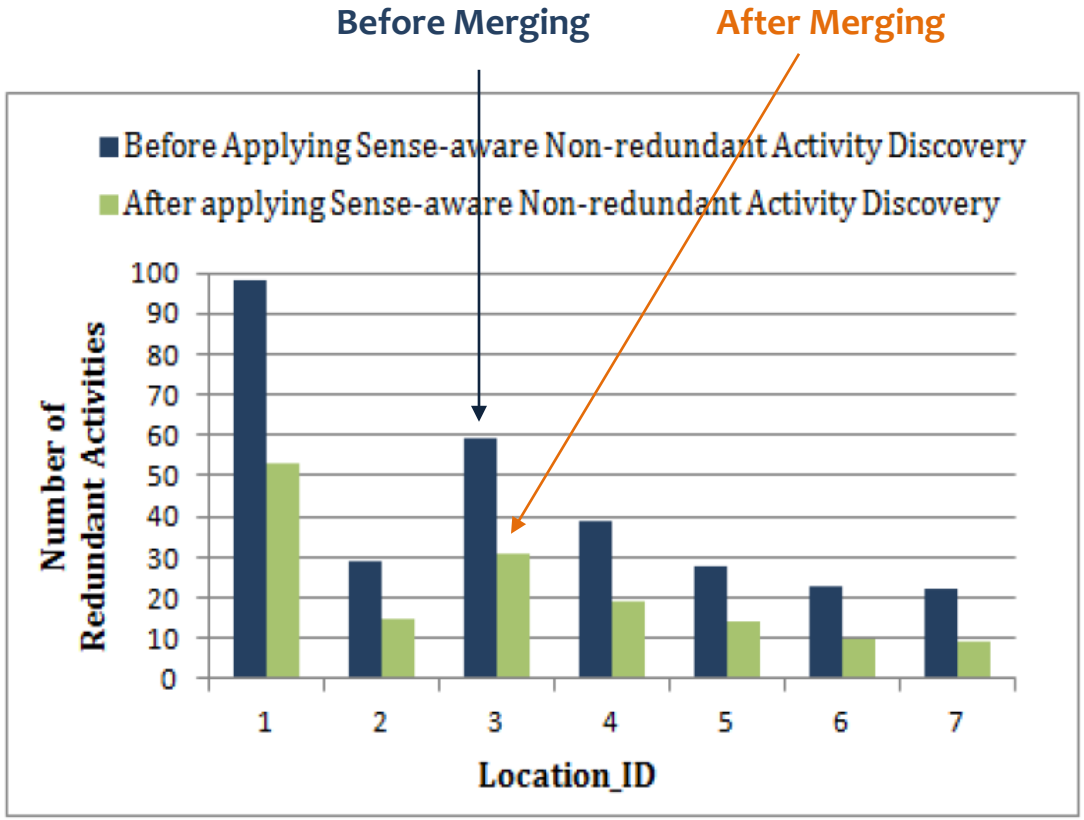
Baseline : 74.78% ActMiner-1 : 83.73%

Qualitative Analysis [Broadcast Service]

Redundancy Minimization on Yelp Dataset

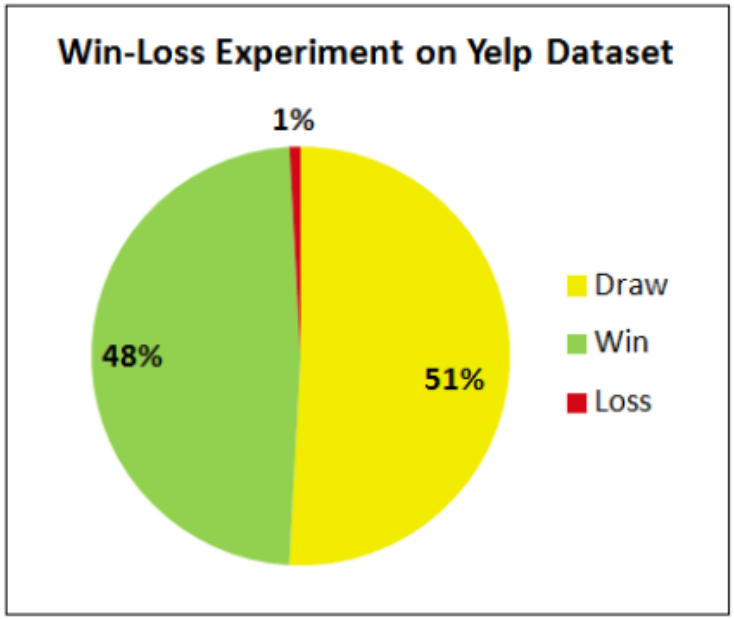
On an Avg. **51.22%** redundancy has been removed in the **Merge Phase** of **ActMiner.**

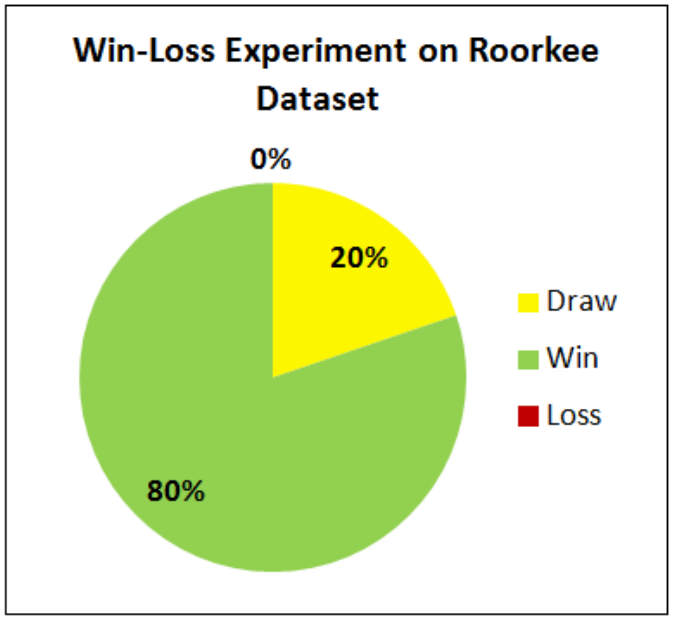
So, now we can push more unique information from the recommender system to the user in broadcast environment.



Qualitative Analysis [Recommendation Performance]

We **have compared ActMiner with Baseline** to evaluate the performance of Location-aware Activity Recommendation **using Win-Loss Experiment.**





And, **ActMiner** outperformed **Baseline** in terms of **quality of recommendation!**

Till Now...

We have seen how *location-specific meaningful, relevant and non-redundant activities* are discovered using ActMiner

But,

Discovering knowledge from data does not ensure its efficient usage.

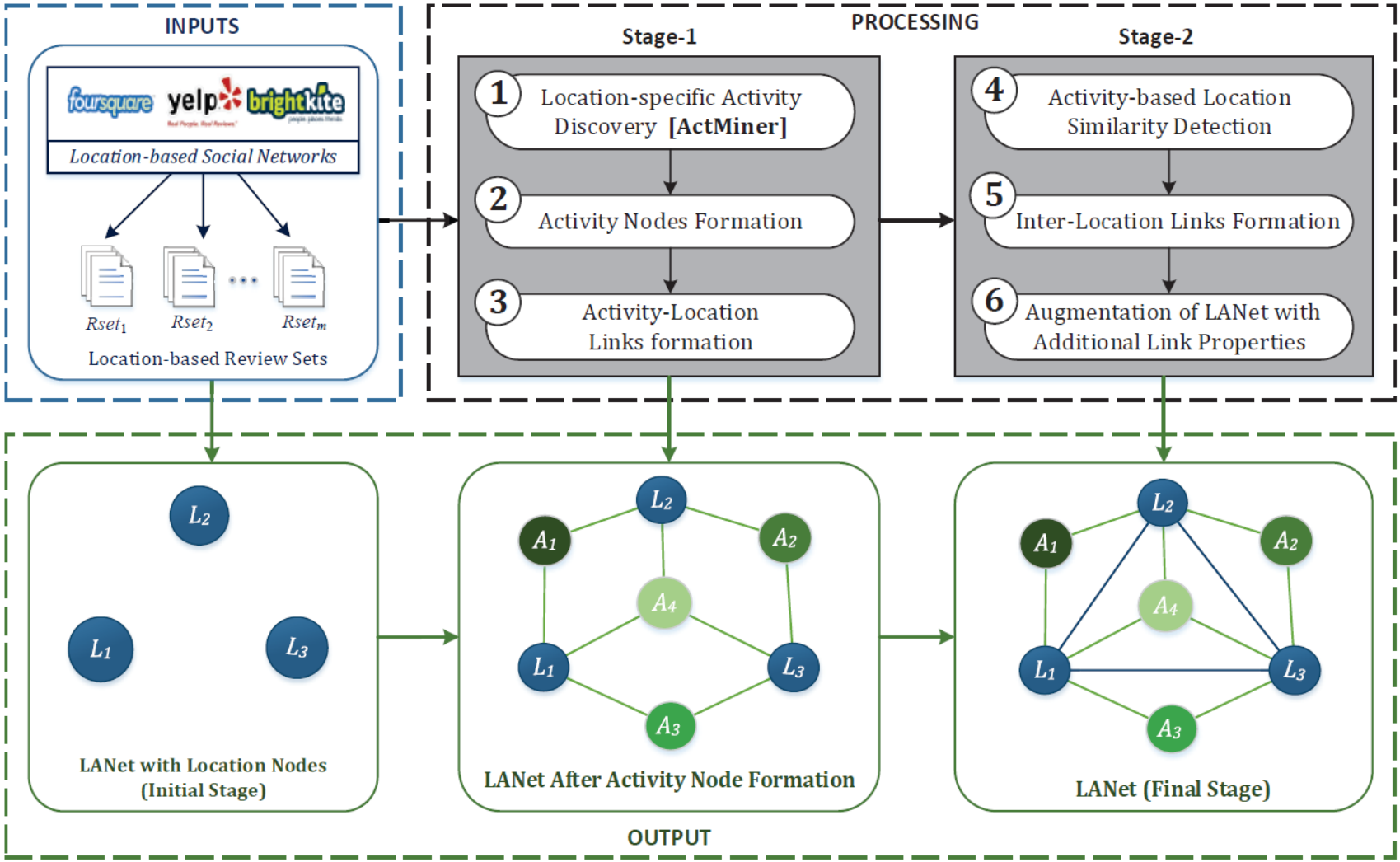
It's the *efficient and systematic way of representing knowledge* that makes it useful.

Now... ➡

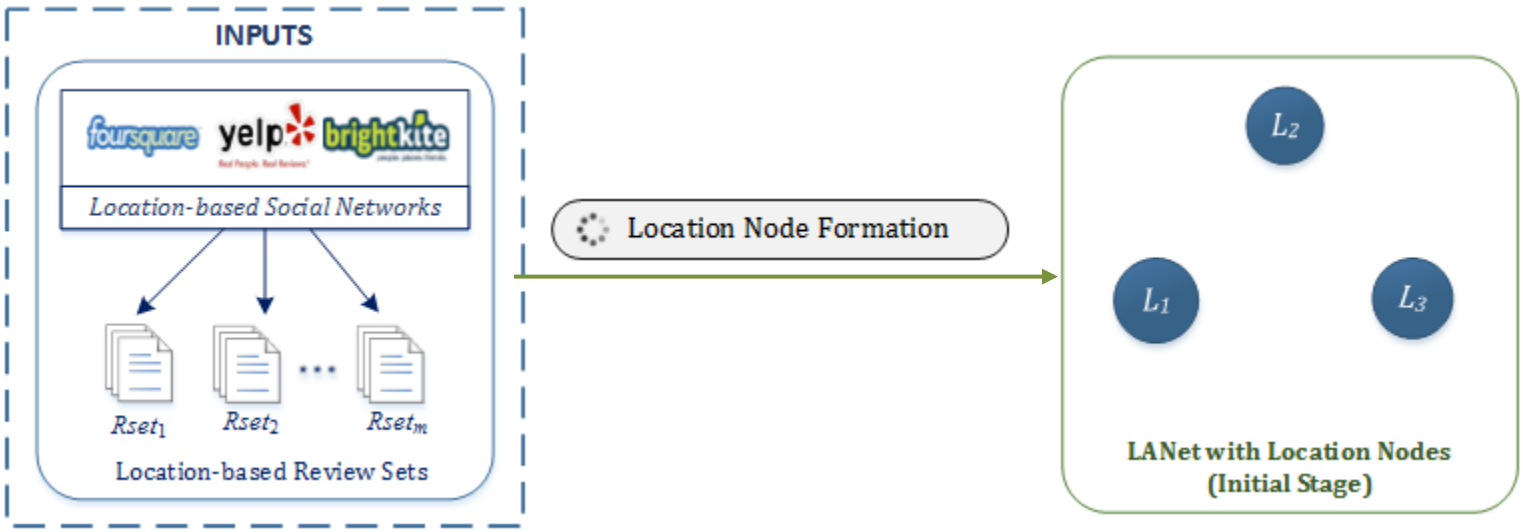
We will construct **LANet -**

The enriched knowledgebase for Location-aware Activity Recommendation System

The LANet Formation Process: An Overview



Initialization Phase: Location Node Formation

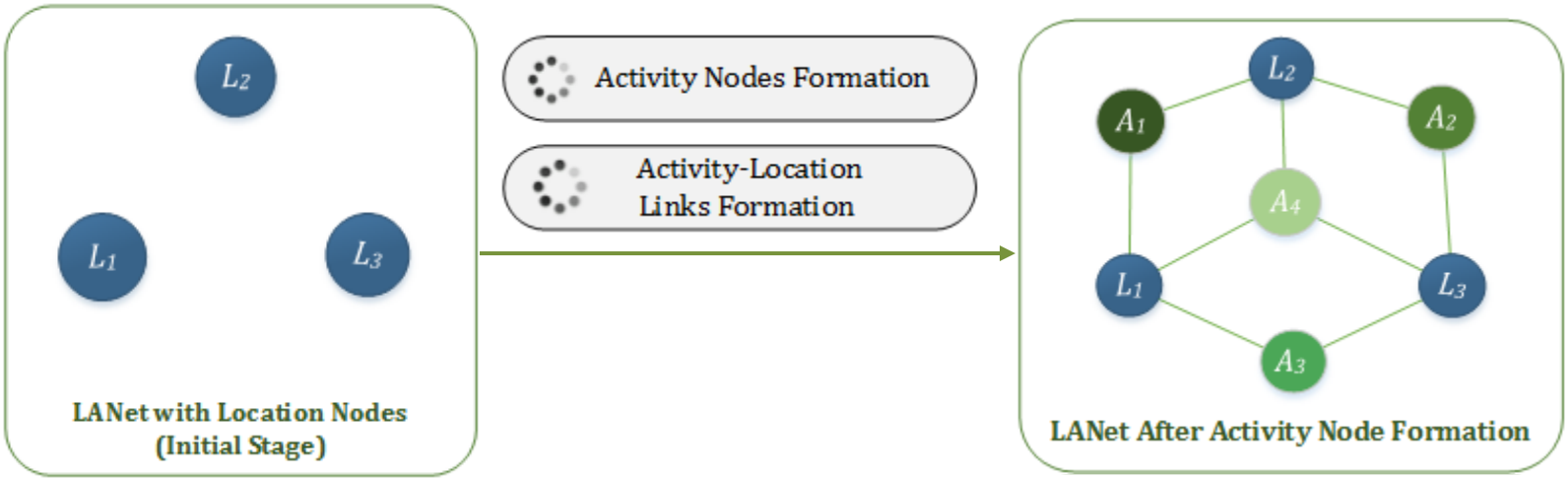


Total ***m* Location Nodes** will be formed for ***m* number of review sets**

Location Node Properties:

Name_of_Location, Formatted Address,
Latitude, Longitude, Category and No_of_Reviews.

Activity Nodes and Activity-Location Links Formation

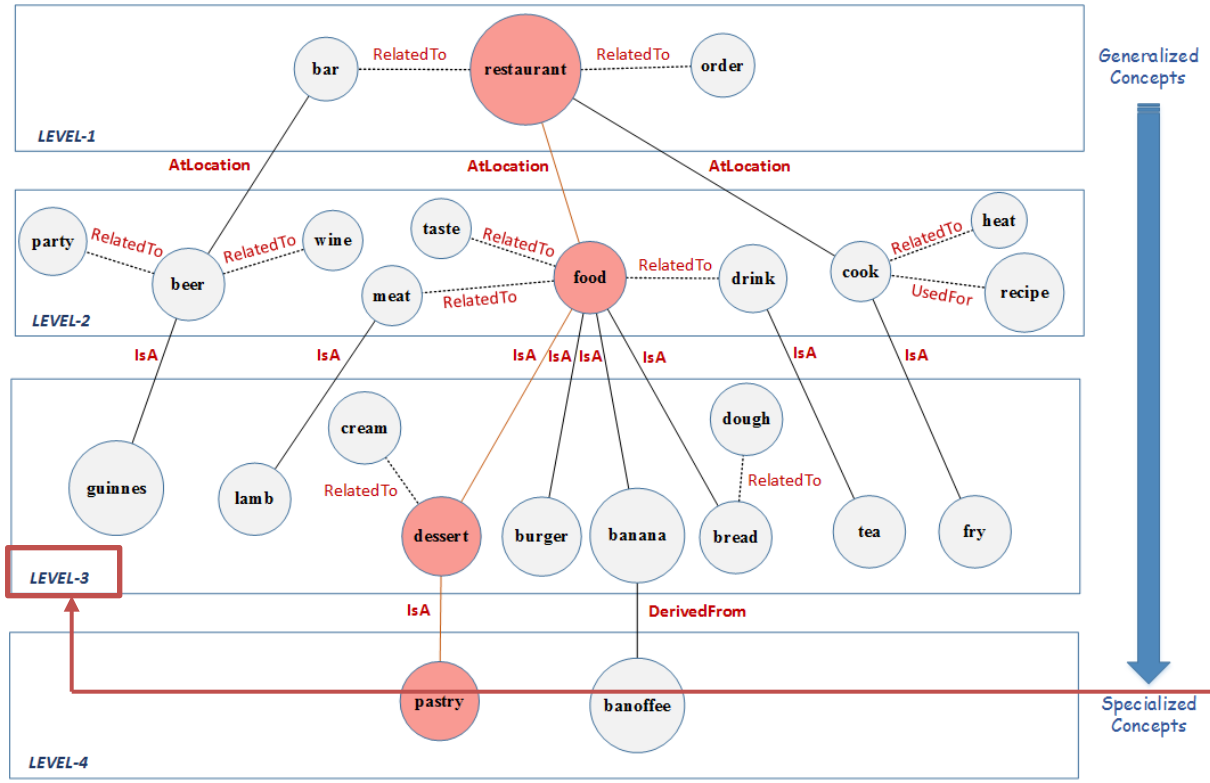


- Inferred activities utilized for **Activity Nodes Formation.**
- Each **Activity Node** are given **Activity_Name** as its property value.
- Next, **Activity Nodes** are connected with **supporting Location Nodes** using **Activity-Location Links** with lebel **"Is_Performed_At"** and Properties:

Actvity_Frequency, Activity_Popularity_Index, Generalized Concept Score, Specialized Concept Score and Boundary_of_Uniqueness.

Calculation of Activity-Location Link Properties

- **Activity_Frequency:** denotes frequency of an activity at a given location and calculated during Activity Discovery using **ActMiner**.
- **Activity_Popularity_Index: LATER!**
- **Generalized_Concept_Score:** used for generalized activity ranking.
- **Specialized_Concept_Score :** used for specilalized activity ranking.
- **Boundary_of_Uniqueness: LATER!**



Generalized Concepts

Specialized Concepts

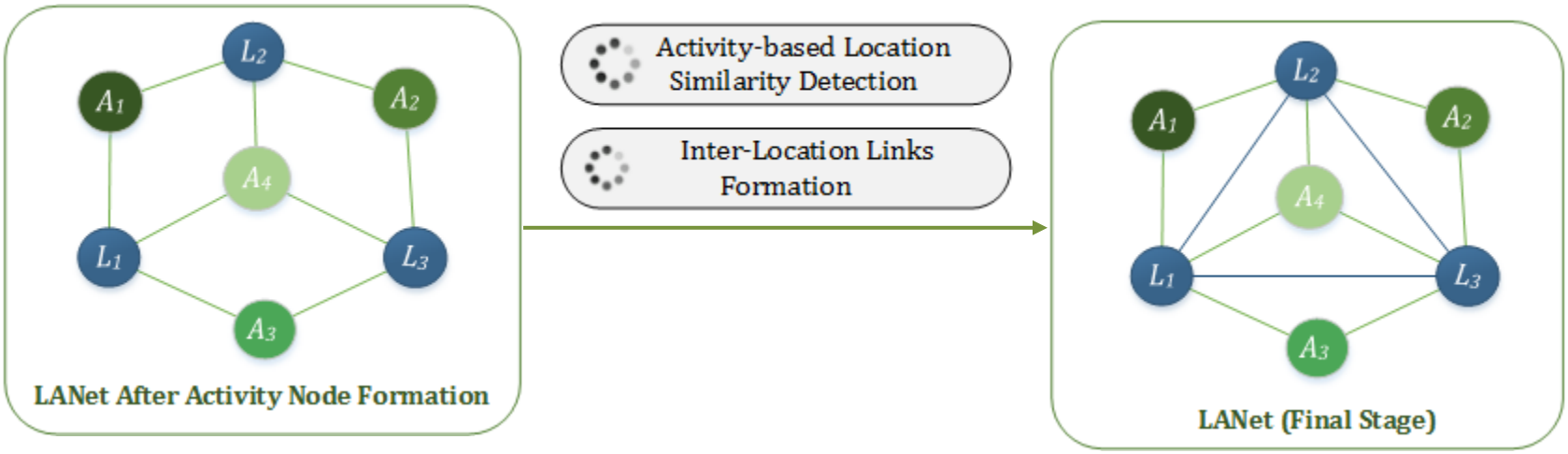
Concept_Frequency

$$GC^k_{score} = \log \boxed{CF_k} \times \frac{1}{CLI_k}$$

$$SC^k_{score} = \log CF_k \times CLI_k$$

Concept_Level_Index

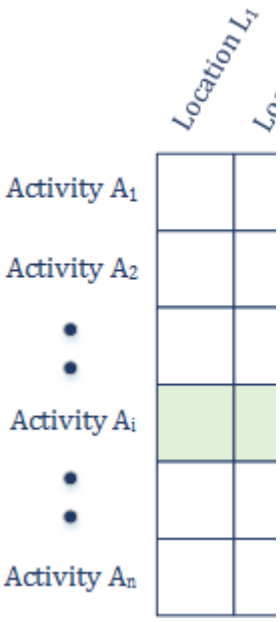
Inter-Location Links Formation

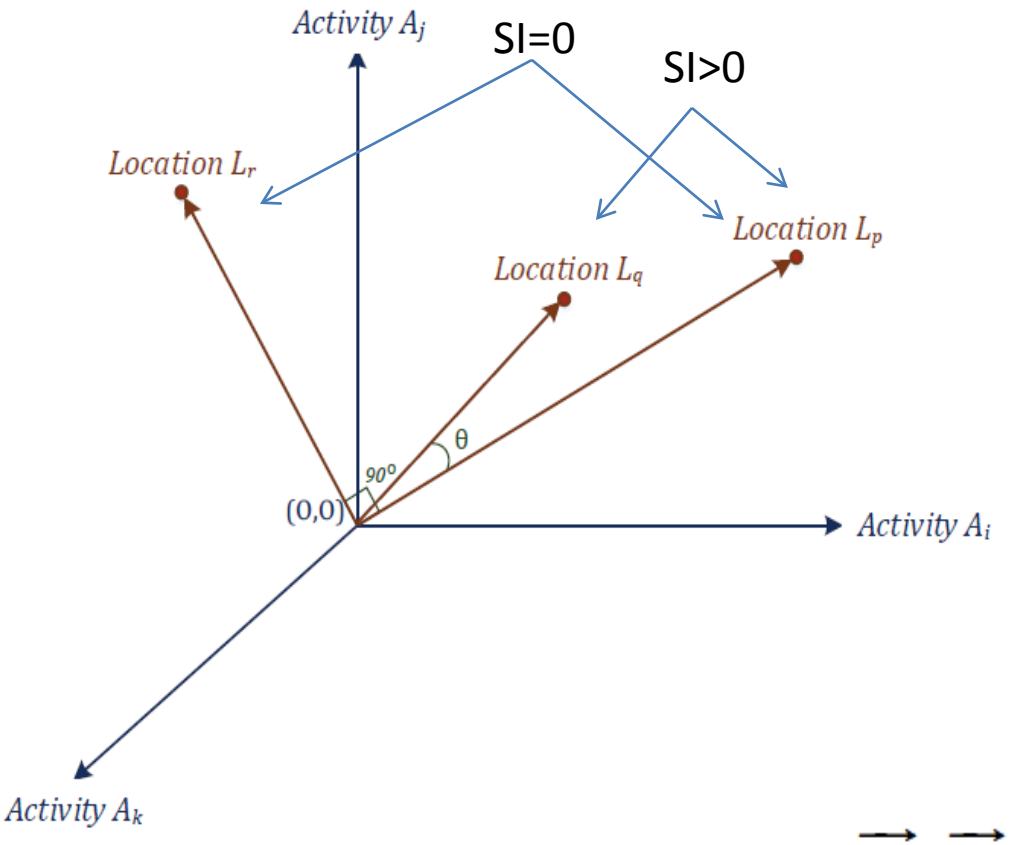


- Infer locations having many **common activities** as **similar locations.**
- **Connect similar location pair** using **Inter-Location Links** with label **"Is_Similar_to"** with following set of properties:

  Similarity_index, Common_Activity_List and Distance.

Location Interpretation of Similarity...





$LM_{ij})$

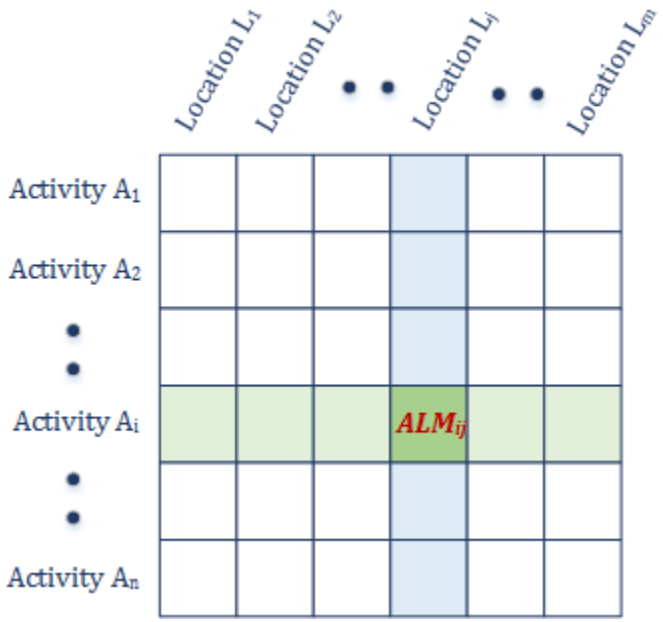
e of an activity
ocation $\mathbf{L_j}$.

$, where 1 \le j \le m$

$\times ILF_i$

$$SI_{pq} = \cos(\overrightarrow{L_p}, \overrightarrow{L_q}) = \frac{\overrightarrow{L_p} \cdot \overrightarrow{L_q}}{|\overrightarrow{L_p}||\overrightarrow{L_q}|}$$

Computation of Activity Popularity Index....

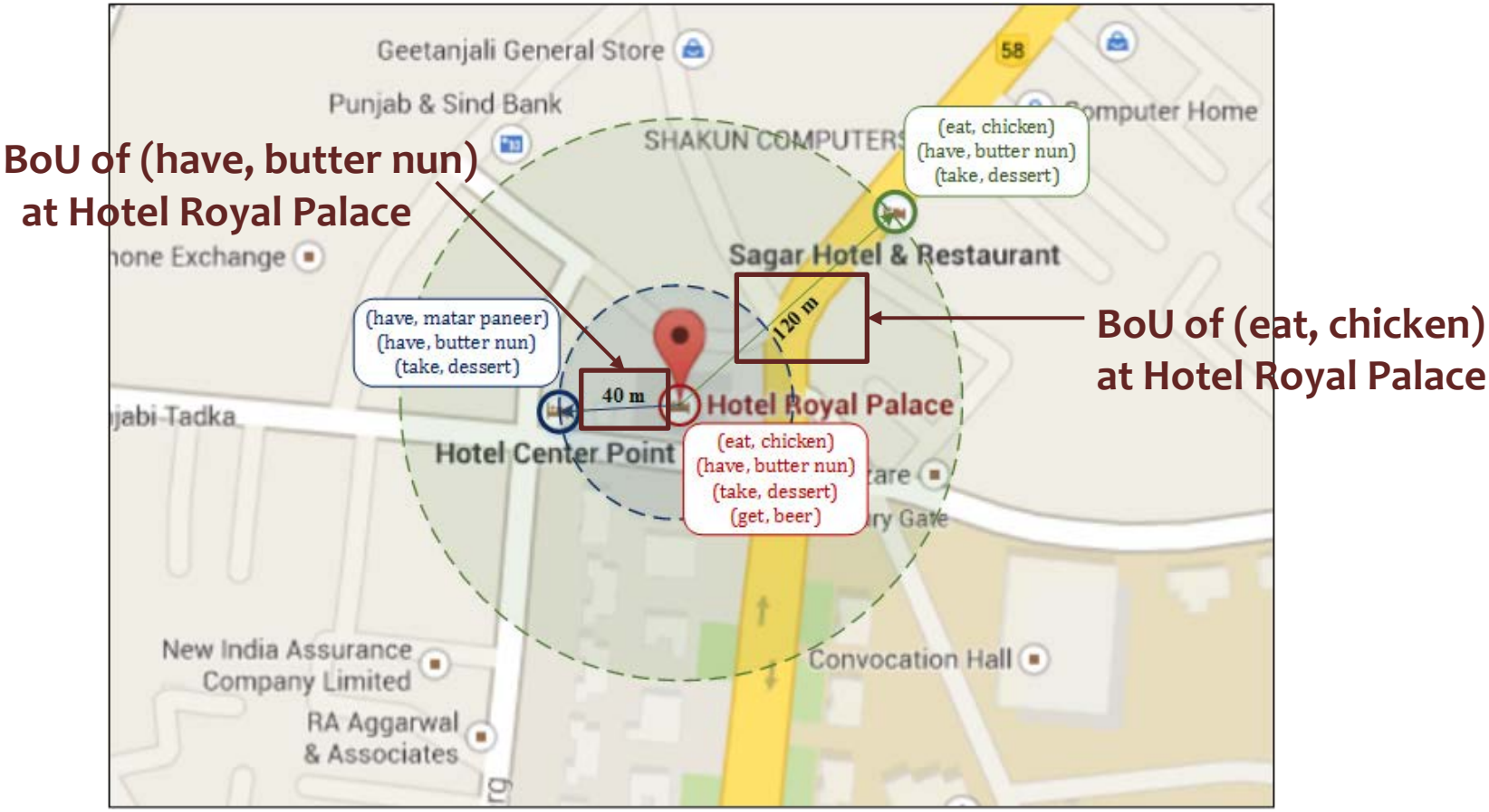


$$API_{ij} = \frac{ALM_{ij}}{\sum_{j=1}^{m} ALM_{ij}} \quad ,where 1 \leq j \leq m$$

After Computing API value for all activities, we augment the **Activity-Location Links** in LANet with these **API** values.

Concept of Boundary of Uniqueness (BoU)....

BoU : **radial distance** of around a given location that covers a **circular area** within which an activity is performed at **no other locations except that location.**



Computation of Boundary of Uniqueness (BoU)....

For calculating BoU value of activities at a given location L_j -

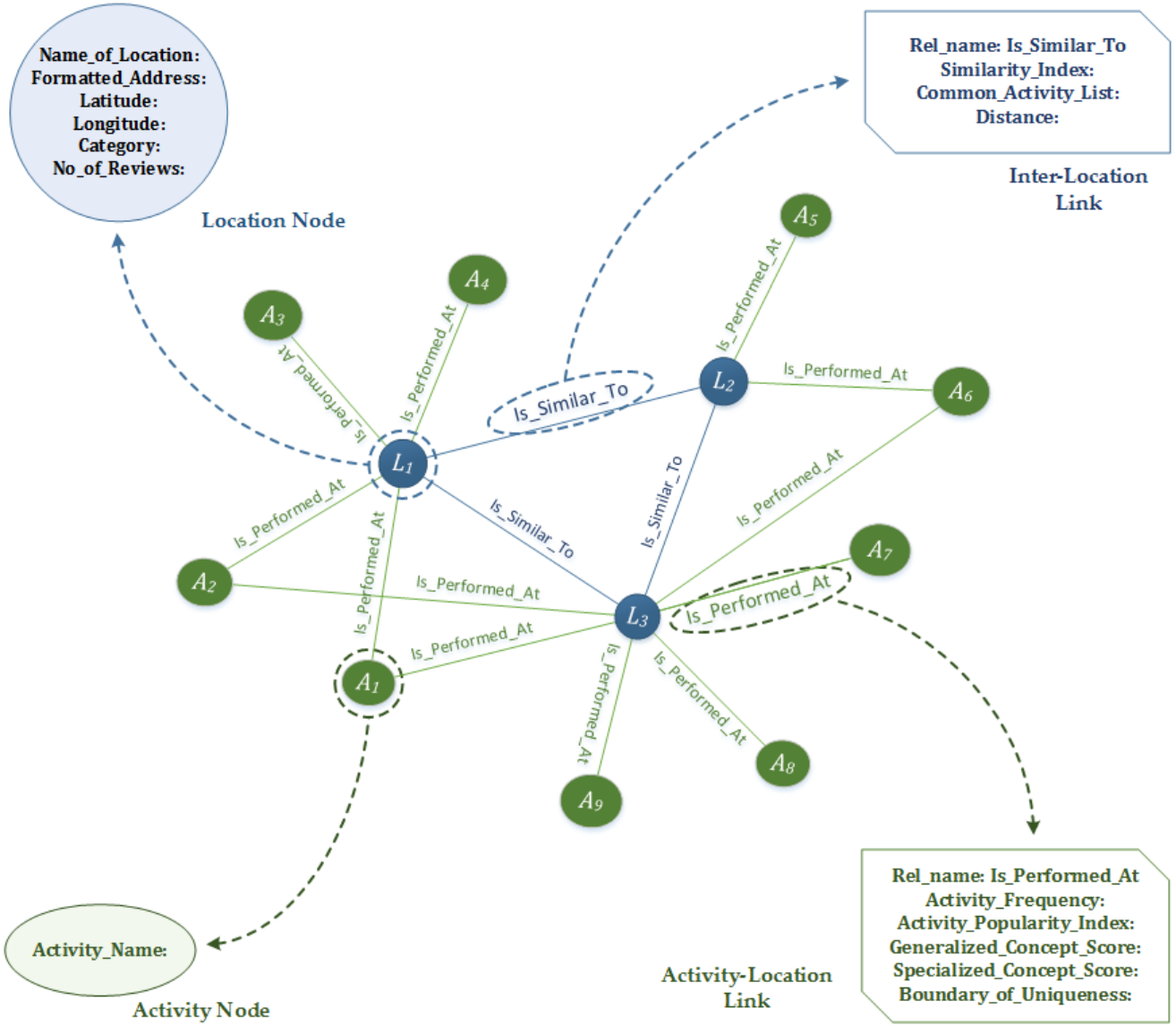
1. Find out Similary_Set of L_j , such that SI(L_j , L_x) > 0 for all $L_x \in L$.
2. Sort Locations in Similary_Set in the increasing order of their distance from

After, BoU calculation phase is over, we augment the **Activity-Location Links** in LANet with these **BoU** values and finally **LANet Discovery Process ends!**

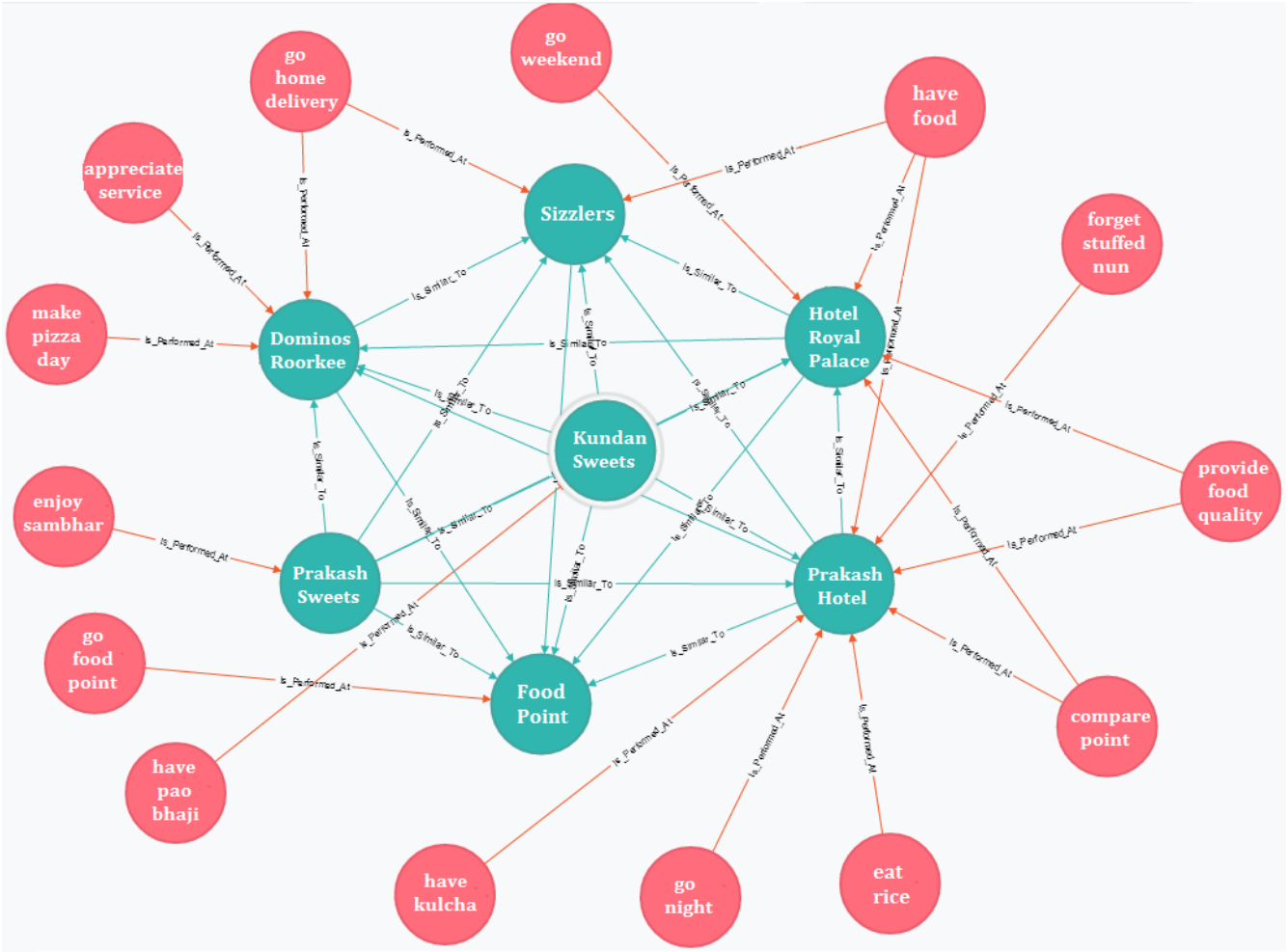
the common activity set to the radial distance of L_j and the nearest location.

5. Iteratively select 2nd , 3rd , … , nth nearest location of L_j and perform step 4 for remaining unassigned activities of L_j .

LANet as a Property Graph ...



Some Statistics of LANet !



Visualizing part of LANet constructed from Roorkee Dataset on Neo4j !

On Roorkee Dataset:

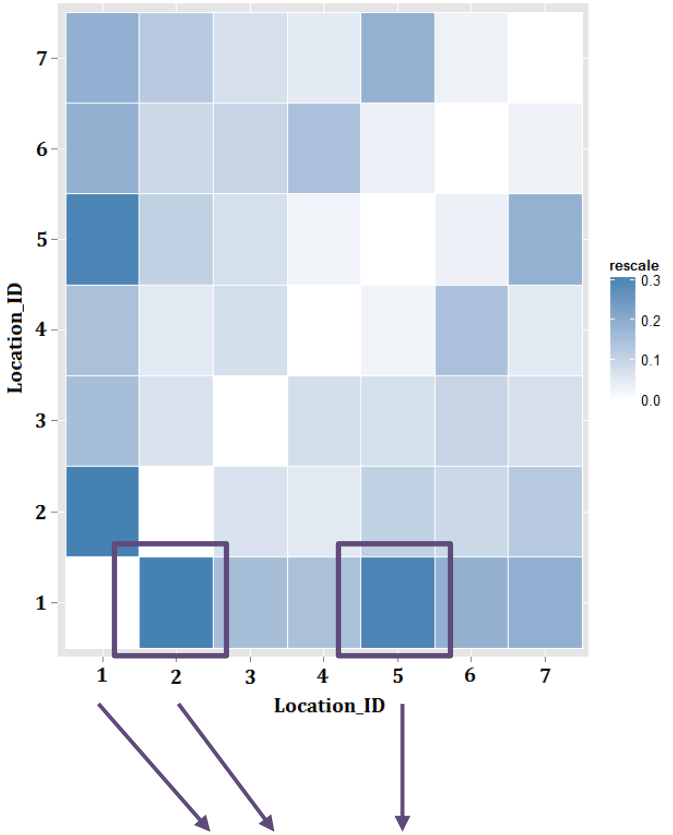
- **25** Location Nodes.
- **886** Activity Nodes.
- **1,077** Activity-Location Links.
- **173** Inter-Location Links.

On Yelp Dataset:

- **7** Location Nodes.
- **6,256** Activity Nodes.
- **7,249** Activity-Location Links.
- **21** Inter-Location Links.

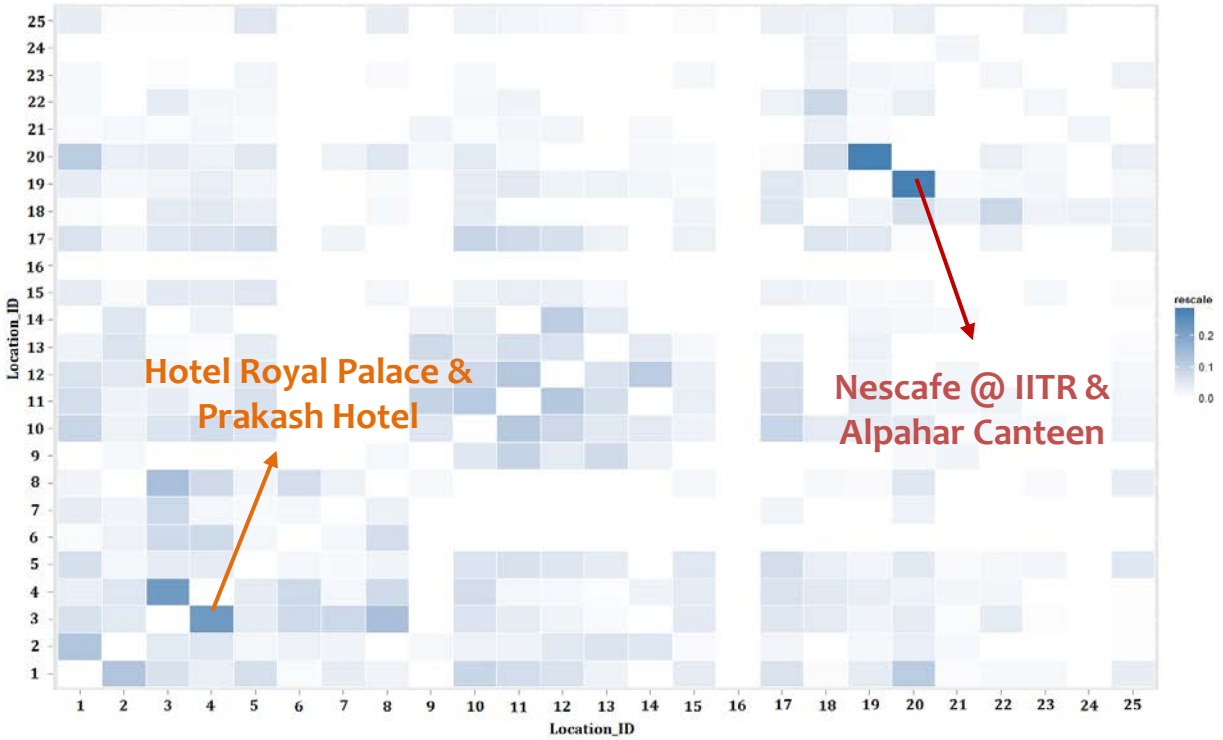
Evaluating Correctness of Information in LANet...

Location Similarity on Yelp



All the 3 locations are restaurants

Location Similarity on Roorkee



Evaluating Correctness of Information in LANet...

Activity Uniqueness

Activities	List of Alternative Locations Ordered with respect to Distance	BoU value	Nearest Alternative Location
(serve, beer)	5, 2, 7	5.24Km	5
(have, experience)	6, 3, 4, 7	4.24Km	6
(have, steak)	2	12.84Km	2
(go, lunch)	5, 2	5.24Km	5
(take, order)	5, 2, 7	5.24Km	5
(have, party)	6, 2, 7	4.24Km	6
(recommend, cheese)	2	12.84Km	2

Cat: Breweries & Restaurant

Uniqueness of 7 activities of LOC_ID-1 of Yelp dataset [Loc: **960 W University Dr Tempe, AZ 85281, USA.**]

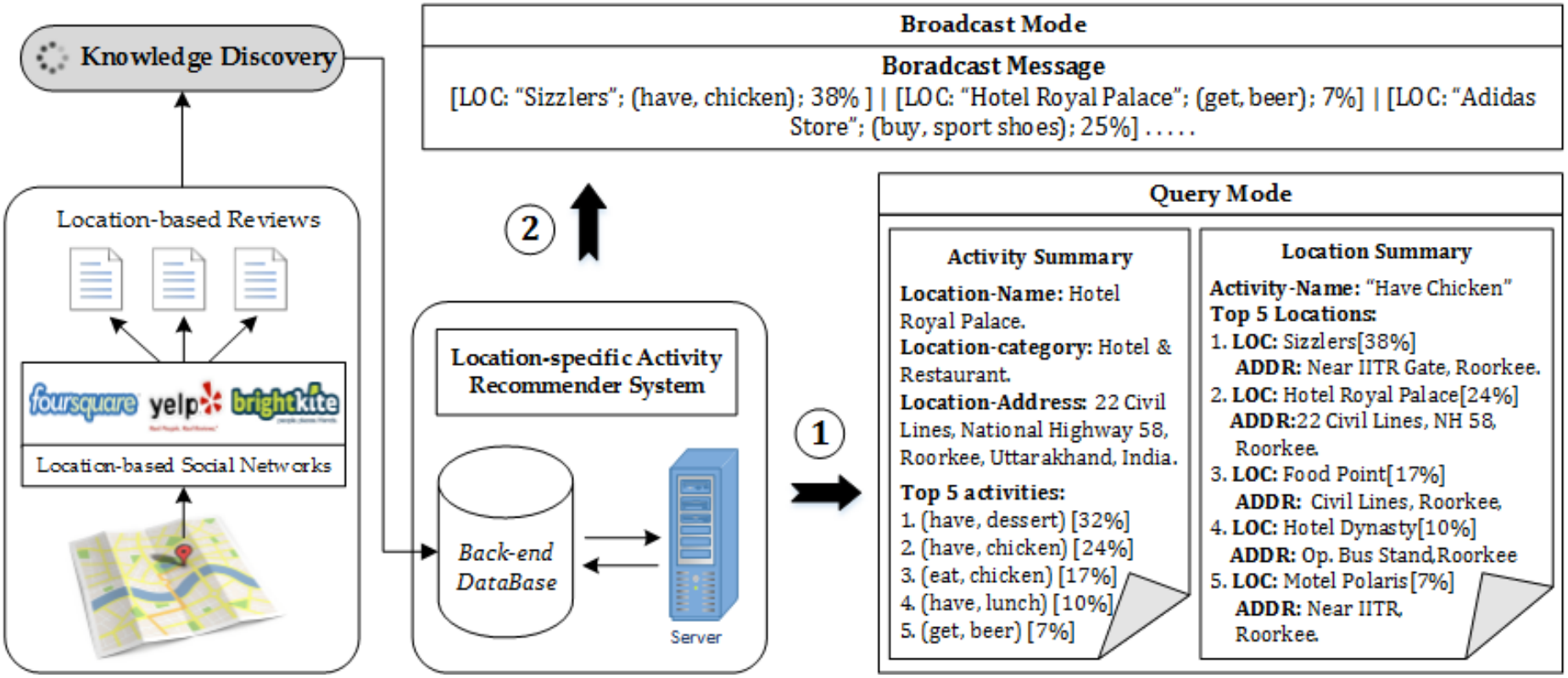
Activities	List of Alternative Locations Ordered with respect to Distance	BoU value	Nearest Alternative Location
(have, dinner)	7, 1, 3, 8, 6	51.45 m	7
(go, friend)	1, 3, 2, 5, 12, 13, 11, 19, 25, 17, 10, 15	77.01 m	1
(serve, food)	3, 2	118.157m	3
(go, birthday party)	3, 8, 6	118.157m	3
(spend, time)	1, 3, 5, 20, 15, 19, 18, 20, 10	77.01 m	1
(have, fun)	3, 18, 17, 10	118.157m	3
(provide, food quality)	3	118.157m	3

Food Point

Uniqueness of 7 activities of LOC_ID-4 of Roorkee dataset [Loc: **Hotel Royal Palace**]

And Finally, multi-mode Recommendation!

A Novel Framework of Location-aware Activity Recommendation System (LActRS) empowered by LANet



Location-aware Review Analytics:

Technical Challenges

Knowledge Discovery: Issues at hand

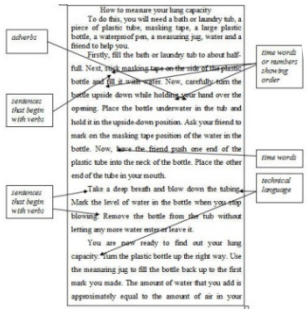
Scalability

- Big Data
- Small Data

Location-aware Reviews

Unstructured

- How to process them to discover knowledge?

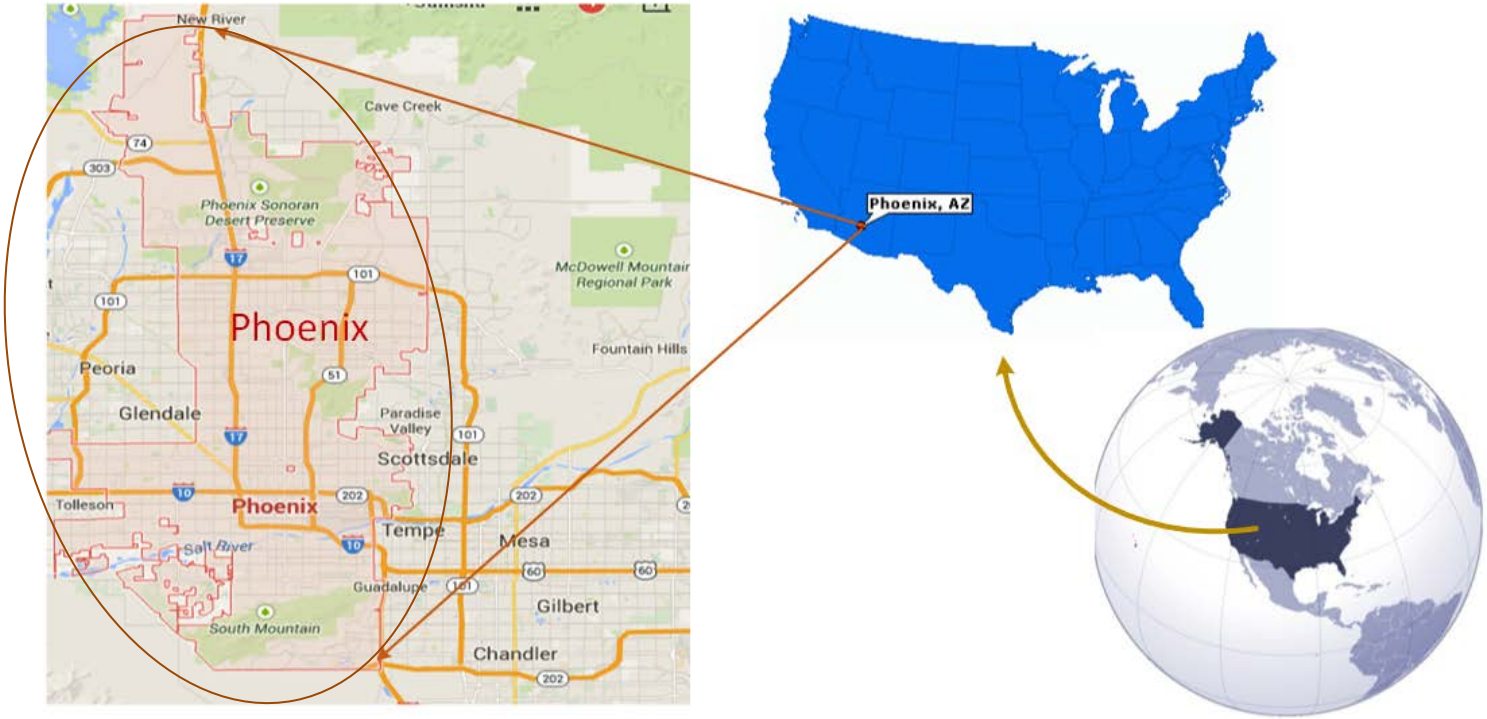


Knowledge Representation

- How to represent large-scale knowledge for efficient storage and retrieval of information?

Challenge I : Scalability

- Recently, Yelp has released a review dataset consisting of **11, 537 business locations in Phoenix, AZ** region and **2,29,907 reviews.**
- *What if we consider millions of such regions with billions of locations across globe and each of them with thousands of reviews on an average?*

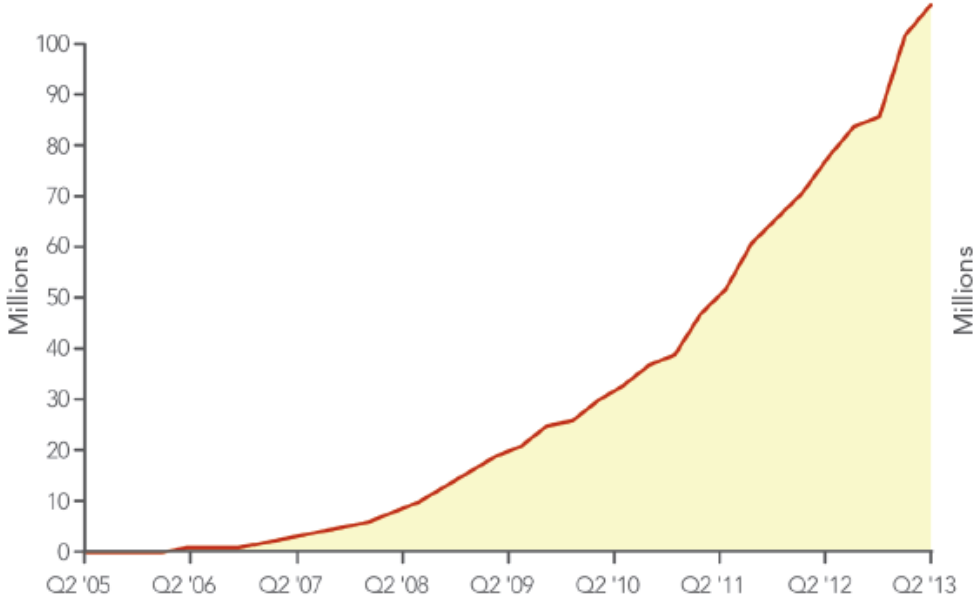


Challenge I : Scalability

*Moreover, the content of location-aware reviews on LBSNs is **diverse**, **sentiment driven** and **growing** day by day which has magnified the scalability problem.*

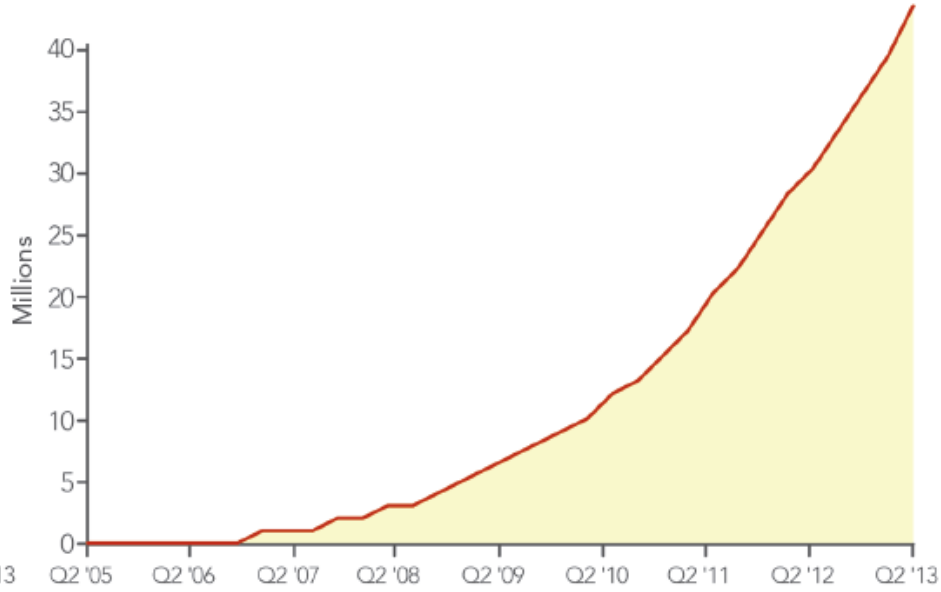
- *As of June 30, 2013, Yelp has gathered **42 million reviews** from **108 million** average monthly unique **visitors**.*

108 Million Monthly Visitors



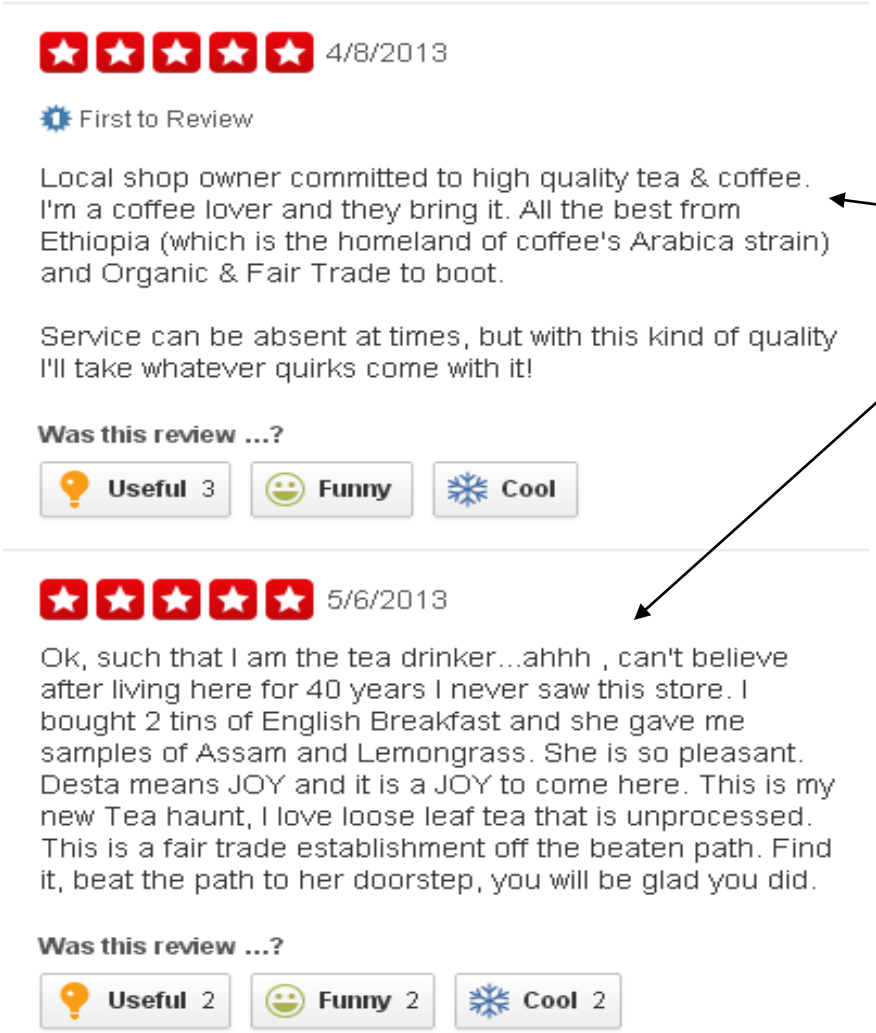
Average monthly unique visitors for the quarter, as measured by Google Analytics

42 Million Reviews



Cumulative reviews contributed since inception

Challenge II : Processing Unstructured Data



- *Location-aware reviews are **unstructured** by nature. How to deal with unstructured data for efficient knowledge discovery ?*
- *What makes the Processing difficult are-*
  - *User generated content.*
  - *Noise.*
  - *Multilingual Information.*
  - *Ambiguity.*
  - *Redundancy.*

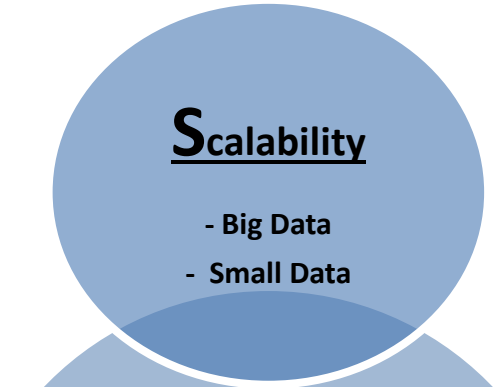
Technical Challenge III : knowledge Representation

How can we represent large-scale and dynamic knowledge such that query processing becomes efficient?

- *The knowledge base should support real-time decision making facility in order to meet the demand of online location-aware recommendation system.*

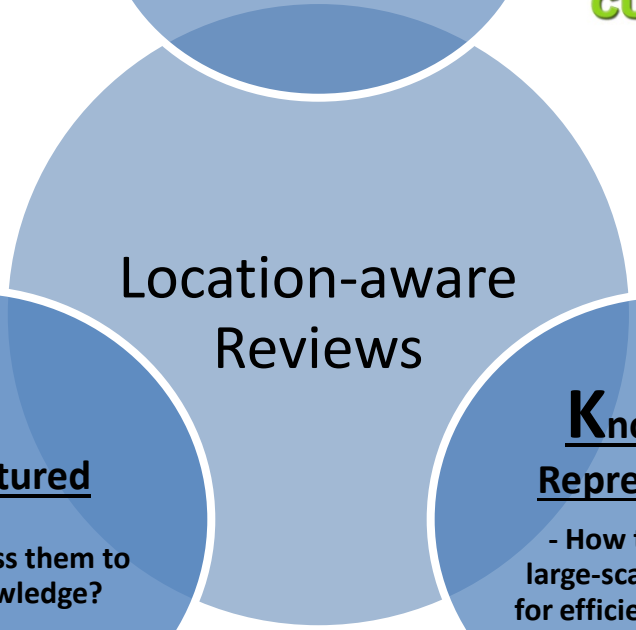
- *Consider a situation - a person is searching for a nearby restaurant for having dessert while moving in a car and when the recommender system responded the location information, he has already left that place behind!*

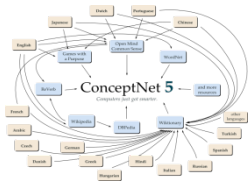
Overcoming Challenges: Solutions

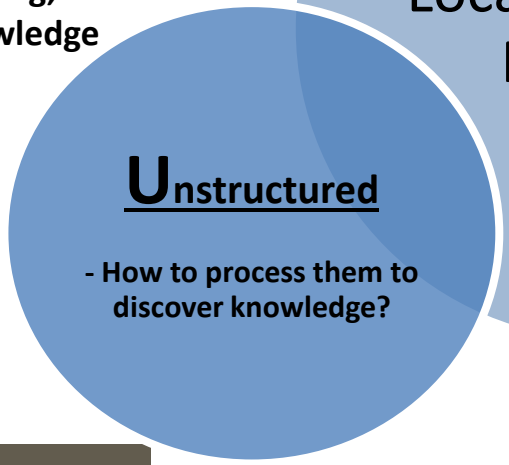


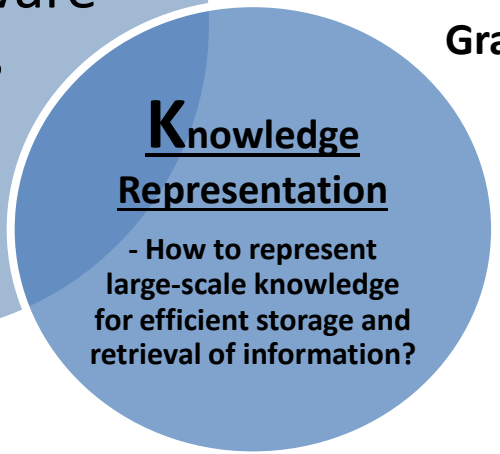
Distributed Computing , Parallel Programming

Natural Language processing, Information retrieval Technique, Machine learning, Semantic Knowledge discovery etc.









Graph Databases

NLP Techniques and Big Data Analytics Platforms:

Overcoming Challenges

We will discuss here …

NLP Techniques for Text Analytics

- Sentence Extraction
- Part-Of-Speech (POS) Tagging
- Named Entity Recognition (NER)
- Dependency Parsing

Semantic Knowledge Discovery of Words

- WordNet
- ConceptNet

Basics of Apache Hadoop FrameWork/ Map-reduce Programming

- Overview of Hadoop Framework
- Map-reduce programming on Hadoop

Sentence Extraction

Objective is to **identify and extract sentences** present in a text corpus.

```
Pierre Vinken, 61 years old, will join the board as a nonexecutive director
Nov. 29. Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.
Rudolph Agnew, 55 years old and former chairman of Consolidated Gold Fields
PLC, was named a director of this British industrial conglomerate.
```

After Sentence Extraction

```
1. Pierre Vinken, 61 years old, will join the board as a nonexecutive
   director Nov. 29.
2. Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.
3. Rudolph Agnew, 55 years old and former chairman of Consolidated Gold
   Fields PLC, was named a director of this British industrial
   conglomerate.
```

Apache OpenNLP Sentence Detector

- The **OpenNLP Sentence Detector** can detect that a punctuation character marks the end of a sentence or not.
- A **sentence** is defined as the *longest white space trimmed character sequence* between two punctuation marks.

```
InputStream modelIn = new FileInputStream("en-sent.bin");
try {
  SentenceModel model = new SentenceModel(modelIn);
}
catch (IOException e) {
  e.printStackTrace();
}
```

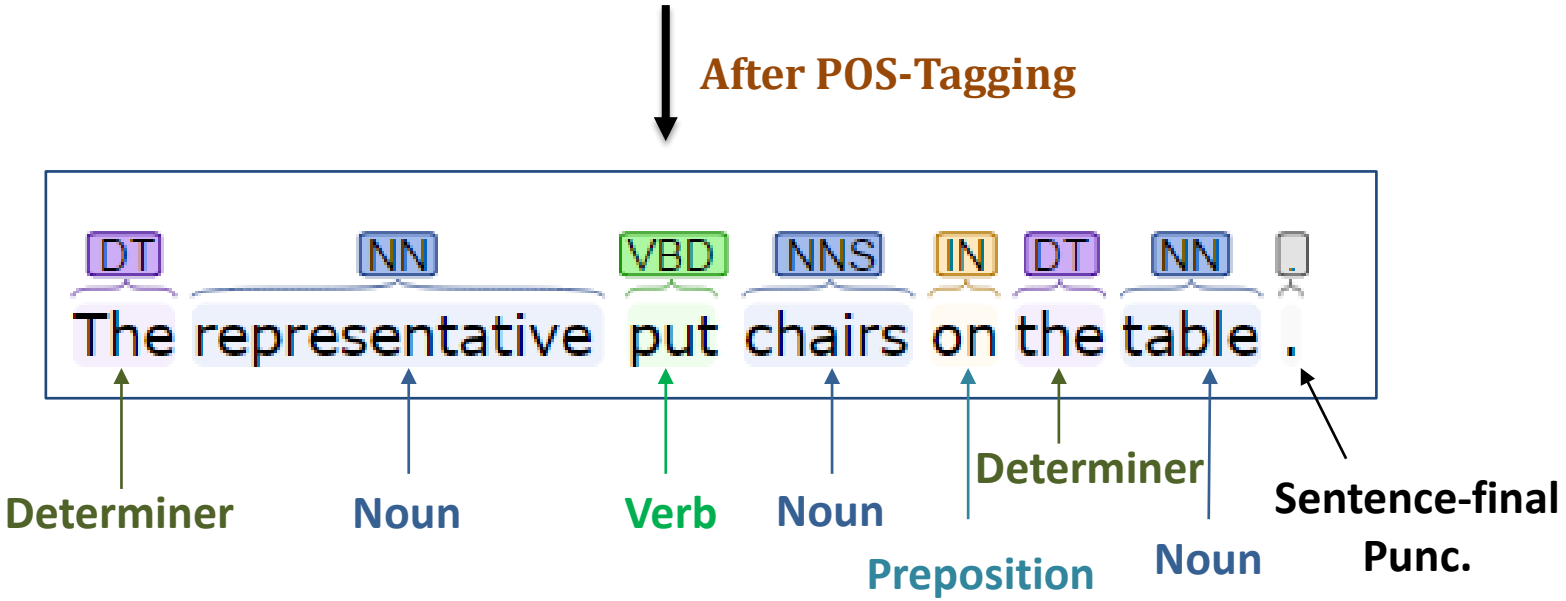
```
SentenceDetectorME sentenceDetector = new SentenceDetectorME(model);
```

```
String sentences[] = sentenceDetector.sentDetect(" First sentence. Second
sentence. ");
```

Parts-of-Speech (POS) Tagging

Objective is to **tag each word (and other token)** of a sentence with its corresponding **parts of speech.**

The representative put chairs on the table .



Penn's Tree Bank Notation

Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*
CD	Cardinal number	*one, two, three*
DT	Determiner	*a, the*
EX	Existential 'there'	*there*
FW	Foreign word	*mea culpa*
IN	Preposition/sub-conj	*of, in, by*
JJ	Adjective	*yellow*
JJR	Adj., comparative	*bigger*
JJS	Adj., superlative	*wildest*
LS	List item marker	*1, 2, One*
MD	Modal	*can, should*
NN	Noun, sing. or mass	*llama*
NNS	Noun, plural	*llamas*
NNP	Proper noun, singular	*IBM*
NNPS	Proper noun, plural	*Carolinas*
PDT	Predeterminer	*all, both*
POS	Possessive ending	*'s*
PP	Personal pronoun	*I, you, he*
PP$	Possessive pronoun	*your, one's*
RB	Adverb	*quickly, never*
RBR	Adverb, comparative	*faster*
RBS	Adverb, superlative	*fastest*
RP	Particle	*up, off*

Tag	Description	Example
SYM	Symbol	+,%, &
TO	"to"	*to*
UH	Interjection	*ah, oops*
VB	Verb, base form	*eat*
VBD	Verb, past tense	*ate*
VBG	Verb, gerund	*eating*
VBN	Verb, past participle	*eaten*
VBP	Verb, non-3sg pres	*eat*
VBZ	Verb, 3sg pres	*eats*
WDT	Wh-determiner	*which, that*
WP	Wh-pronoun	*what, who*
WP$	Possessive wh-	*whose*
WRB	Wh-adverb	*how, where*
$	Dollar sign	*$*
#	Pound sign	*#*
"	Left quote	(' or ")
"	Right quote	(' or ")
(	Left parenthesis	([, (, {, <)
)	Right parenthesis	( ],), }, >)
,	Comma	,
.	Sentence-final punc	(. ! ?)
:	Mid-sentence punc	(: ; ... – -)

Named Entity Recognition

Objective is to ***Identify mentions in text and classify them into a predefined set of categories of interest.***

Prof. Jerry Hobbs taught CS544 during February 2010. Hobbs corporation bought FbK.

After Named Entity Recognition

<PER>**Prof. Jerry Hobbs**</PER> taught CS544 during <DATE>**February 2010**</DATE>.
<ORG>**Hobbs corporation**</ORG> bought <ORG>**FbK**</ORG>.

Predefined set of categories of interest:

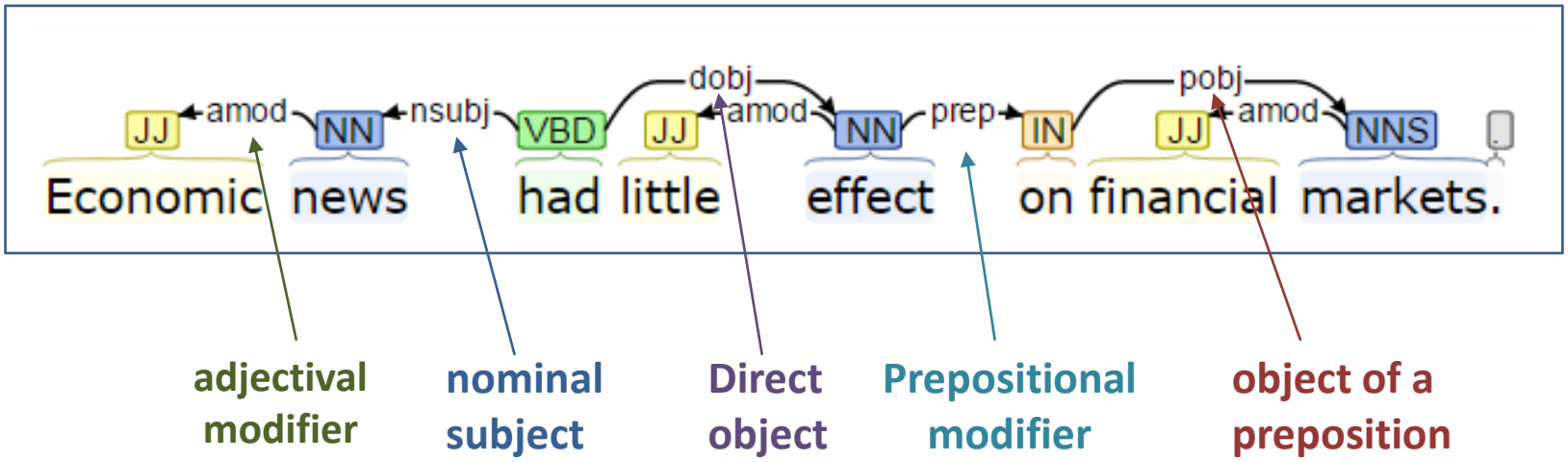
Person Names: **Prof. Jerry Hobbs**
Organizations: **Hobbs corporation**, **FbK**
Date and time expressions: **February 2010**

Dependency Parsing

Objective is to provide a simple description of the ***grammatical relationships in a sentence*** that can easily be understood and effectively used by people **without linguistic expertise who want to extract textual relations.**

Economic news had little effect on financial markets.

After Dependency Parsing



Types of Dependency Parsing

Economic news had little effect on financial markets.

Typed dependencies

nn(news-2, Economic-1)
nsubj(had-3, news-2)
root(ROOT-0, had-3)
amod(effect-5, little-4)
dobj(had-3, effect-5)
prep(effect-5, on-6)
amod(markets-8, financial-7)
pobj(on-6, markets-8)

Typed dependencies, collapsed

nn(news-2, Economic-1)
nsubj(had-3, news-2)
root(ROOT-0, had-3)
amod(effect-5, little-4)
dobj(had-3, effect-5)
amod(markets-8, financial-7)
prep_on(effect-5, markets-8)

Stanford Typed Dependencies

- The current representation contains approximately 50 grammatical relations.
- Dependencies are all binary relations:
  - a grammatical relation holds between a governor (also known as a regent or a head) and a dependent.
- Most generic grammatical relation, **dependent (dep)**, will be used when a more precise relation in the hierarchy does not exist or cannot be retrieved by the system.

Further Reading:

Stanford typed dependencies manual
Marie-Catherine de Marnee and Christopher D. Manning

Source: http://robotics.usc.edu/~gkoch/DependencyManual.pdf

Semantic Knowledge Discovery of Words:

WordNet

&

ConceptNet

WordNet 3.0

- A hierarchically organized lexical database.
- On-line thesaurus + aspects of a dictionary.
- Some other languages available or under development
  - (Arabic, Finnish, German, Portuguese, Hindi, …)

Category	Unique Strings
Noun	117,798
Verb	11,529
Adjective	22,479
Adverb	4,481

WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: eat Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Verb

- S: (v) **eat** (take in solid food) *"She was eating a banana"; "What did you eat for dinner last night?"*
- S: (v) **eat** (eat a meal; take a meal) *"We did not eat until 10 P.M. because there were so many phone calls"; "I didn't eat yet, so I gladly accept your invitation"*
- S: (v) feed, **eat** (take in food; used of animals only) *"This dog doesn't eat certain kinds of meat"; "What do whales eat?"*
- S: (v) **eat**, eat on (worry or cause anxiety in a persistent way) *"What's eating you?"*
- S: (v) consume, eat up, use up, **eat**, deplete, exhaust, run through, wipe out (use up (resources or materials)) *"this car consumes a lot of gas"; "We exhausted our savings"; "They run through 20 bottles of wine a week"*
- S: (v) corrode, **eat**, rust (cause to deteriorate due to the action of water, air, or an acid) *"The acid corroded the metal"; "The steady dripping of water rusted the metal stopper in the sink"*

How is "Sense" defined in WordNet ?

- The **synset (synonym set)**, the set of near-synonyms, instantiates a sense or concept with a gloss.
- Example: **chump** as a noun with the **gloss:**

  "a person who is gullible and easy to take advantage of"

  - **chump[1], fool[2], gull[1], mark[9], patsy[1], fall guy[1], sucker[1], soft touch[1], mug[2]**

  - Each of these senses have this same gloss
  - (Not every sense; sense 2 of gull is the aquatic bird)

WordNet Hypernym Hierarchy for "bass"

Noun

- S: (n) **bass** (the lowest part of the musical range)
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) pitch (the property of sound that varies with variation in the frequency of vibration)
      - S: (n) sound property (an attribute of sound)
        - S: (n) property (a basic or essential attribute shared by all members of a class) *"a study of the physical properties of atomic particles"*
          - S: (n) attribute (an abstraction belonging to or characteristic of an entity)
            - S: (n) abstraction, abstract entity (a general concept formed by extracting common features from specific examples)
              - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

WordNet Noun Relations

Relation	Also Called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	Breakfast[1] -> Meal[1]
Hyponym	Subordinate	From concepts to subtypes	Meal[1] -> Lunch[1]
Member Meronym	Has-Member	From Groups to their members	Faculty[2] -> Professor[1]
Has-Instance		From concepts to instances of the concepts	Composer[1] -> Bach[1]
Instance		From instances to their concepts	Austen[1] -> Author[1]
Member Holonym	Member-Of	From members to their groups	Copilor[1] -> Crew[1]
Part Meronym	Has-Part	From wholes to parts	Table[2] -> Leg[3]
Part Holonym	Part-Of	From Parts to wholes	Course[7] -> Meal[1]
Antonym		opposites	Leader[1] -> follower[1]

WordNet 3.0

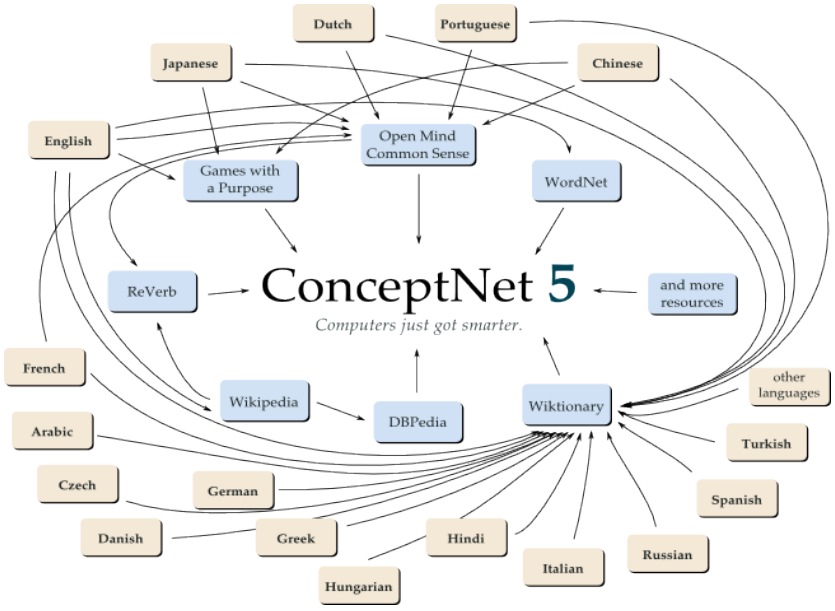
- Where it is:
  http://wordnetweb.princeton.edu/perl/webwn

- Libraries
  - Python : WordNet from NLTK
    http://www.nltk.org/howto/wordnet.html

  - Java: JWNL (Java WordNet Library),
    JAWS (Java API for WordNet Searching)

ConceptNet: What is it?

- A **crowd sourced knowledgebase of common sense.** Basically, it's a semantic network containing lots of things computers should know about the world, especially when understanding text written by people.
- Built from **nodes** representing **words or short phrases** of natural language, and **labeled relationships between them.**
- **Application:** To search for information better, answer questions, and understand people's goals.
- Created by contributors to the **Open Mind Common Sense project,** from **MIT Media Lab.**



Sources of Knowledge in ConceptNet 5

- ConceptNet 5 contains almost all the data from **ConceptNet 4**.
- A subset of **DBPedia**, which *extracts knowledge from the infoboxes on Wikipedia articles.*
- **Wiktionary**, the free multilingual dictionary, a sister project to Wikipedia.
  [source of information about synonyms, antonyms, translations of concepts into hundreds of languages, and multiple labeled word senses for many words.]
- More dictionary-style knowledge comes from **WordNet**.
- UMBEL connects ConceptNet to the **OpenCyc ontology** via a Semantic Web representation.
- People's intuitive word associations comes from "**games with a purpose**".

Assertions

Two concepts connected by a relation and justified by sources

dog — *IsA* → mammal
a dog is a kind of mammal

cat — *Antonym* → dog
cat is not dog

run — *HasSubevent* → sweat
One of the things you do when you run is sweat

human — *CapableOf* → run
Humans can run

brown — *InstanceOf* → poet

不敵 — *TranslationOf* → dare
不敵 is Japanese for daring

Types of Relations

- IsA
- UsedFor
- RelatedTo
- AtLocation
- HasA
- DefinedAs
- CreatedBy
- HasProperty
- DerivedFrom
- MotivationOf
- Desires
- MadeOf
- CapableOf
- TranslationOf
- InheritsFrom
- LocatedNear
- Synonym
- Antonym
- ConceptuallyRelatedTo
- EffectOf

Structure

Concepts

- ✓ Words and phrases
- ✓ Represented as nodes

Predicates

- ✓ Set of relations such as IsA or HasA
- ✓ Also represented as nodes

Assertions

- ✓ Represented by edges in the graph that connect multiple nodes in the graph (concepts and relations)
- ✓ An edge is an instance of an assertion.

Using ConceptNet

- JSON REST API
- Running it locally
  - Can choose which data to use and add more data
- All the data is available for convenience
  - Flat JSON file
  - Solr JSON file
  - CSV

Web Frontend: http://conceptnet5.media.mit.edu/

Apache Hadoop Framework and Map-reduce Programming

The Need for "Distributed Processing"

- Data is everywhere and growing exponentially.
  - Accumulation of data on web.
  - Machines, too, are generating and keeping more and more data.

- **Challenge:** Need to go through **terabytes and petabytes of data** to figure out
  - *Which websites were popular?*
  - *What books were in demand ?*
  - *What kinds of ads appealed to people?* ..... And many more!

- But, Large scale data processing was difficult!
  - Managing hundreds or thousands of processors
  - Managing parallelization and distribution
  - I/O Scheduling
  - Status and monitoring
  - Fault/crash tolerance

- **Solution:** A distributed Framework for Processing Large Scale Data : **Hadoop** and **Map-Reduce Programming.**

Apache Hadoop: Overview

- An open source framework for **writing and running distributed applications that process large amounts of data.**

- **Idea:**
  - Tie together many low-end/commodity machines together as a single functional distributed system.
  - Data set is divided into smaller (typically 64 MB) blocks that are spread among many machines in the cluster via the Hadoop Distributed File System (HDFS).
  - With a modest degree of replication, the cluster machines can read the data set in parallel and provide a much higher throughput.
  - focuses on moving code to data instead of vice versa.

- **key distinctions of Hadoop:**
  - **Accessible:** runs on large clusters of commodity machines or cloud computing services (Amazon EC2)
  - **Robust:** gracefully handles fault tolerance.
  - **Scalable:** scales linearly to handle larger data
  - **Simple:** allows users to quickly write efficient parallel code.

Understanding Map-Reduce

- A data processing model .
- **Greatest Advantage:**
  "Easy scaling of data processing over multiple computing nodes."
- MapReduce programs are executed in two main phases, called **mapping** and **reducing.** Each phase is defined by a data processing function, and these functions are called mapper and reducer, respectively.
- In the mapping phase, MapReduce takes the input data and feeds each data element to the mapper.
- In the reducing phase, the reducer processes all the outputs from the mapper and arrives at a final result.
- Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once you write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change.

A simple WordCount program

Objective: To compute the frequency of each word in a text file.

Input Text: *Do as I say, not as I do.*

Pseudo-code for a WordCount Program:

```
define wordCount as Multiset;
for each document in documentSet {
    T = tokenize(document);
    for each token in T {
        wordCount[token]++;
    }
}
display(wordCount);
```

Output

Word	Count
as	2
do	2
i	2
not	1
say	1

But, What if the Input Data is a large set of documents ?

The Map-Reduce Programming Paradigm

- Input must be structured as a list of (key/value) pairs , list(<k1, v1>).
  - Example:
    <file_name, file_content>

	Input	Output
map	<k1, v1>	list(<k2, v2>)
reduce	<k2, list(v2)>	list(<k3, v3>)

- The list of (key/value) pairs is broken up and each individual (key/value) pair, <k1, v1>, is transformed by mapper into a list of <k2, v2> pairs. Example : { <"foo", 1>, <"foo", 1>, <"foo", 1>}.

- The output of all the mappers are (conceptually) aggregated into one giant list of <k2, v2> pairs. All pairs sharing the same k2 are grouped together into a new (key/value) pair, <k2, list(v2)>. Example: <"foo", list(1,1,1,)>.

- The framework asks the reducer to process each one of these aggregated (key/value) pairs individually and finally generates list(<k3, v3>). Example: { <"foo", 3> , ...}.

Scaling WordCount program in Map-Reduce

Pseudo-code for Map and Reduce functions for word counting

```
map(String filename, String document) {
    List<String> T = tokenize(document);
    for each token in T {
        emit ((String)token, (Integer) 1);
    }
}
reduce(String token, List<Integer> values) {
    Integer sum = 0;

    for each value in values {
        sum = sum + value;
    }
    emit ((String)token, (Integer) sum);
}
```

WordCount in Map-Reduce: A Java Implementation

```
public class WordCount extends Configured implements Tool {

  public static class MapClass extends MapReduceBase
    implements Mapper<LongWritable, Text, Text, IntWritable> {

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(LongWritable key, Text value,
                    OutputCollector<Text, IntWritable> output,
                    Reporter reporter) throws IOException {
      String line = value.toString();
      StringTokenizer itr = new StringTokenizer(line);
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        output.collect(word, one);
      }
    }
  }

  public static class Reduce extends MapReduceBase
    implements Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text key, Iterator<IntWritable> values,
                       OutputCollector<Text, IntWritable> output,
                       Reporter reporter) throws IOException {
      int sum = 0;
      while (values.hasNext()) {
        sum += values.next().get();
      }
      output.collect(key, new IntWritable(sum));
    }
  }

  ...
}
```

1. Tokenize using white spaces
2. Cast token into Text object
3. Output count of each token

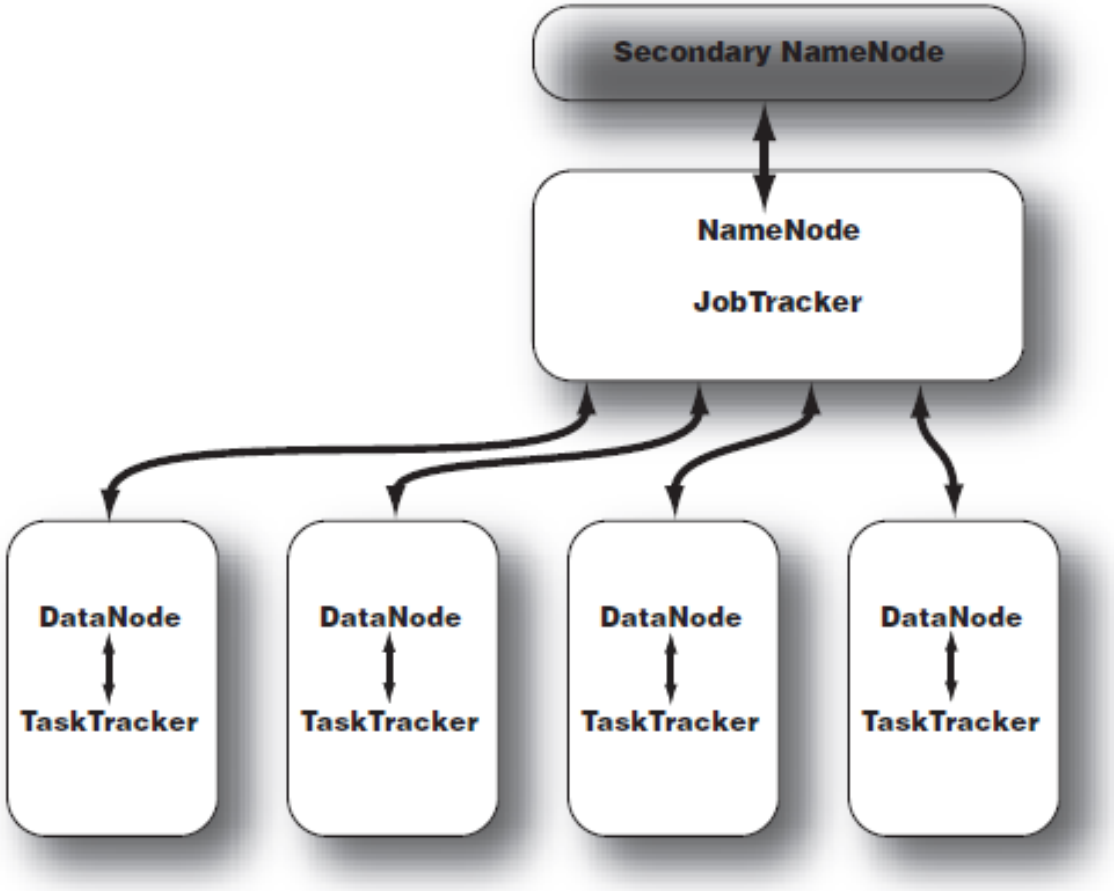
The Hadoop Architecture

❑ On a fully configured cluster, "**running Hadoop**" means **running a set of daemons**, or **resident programs**, on the different servers in your network.

❑ The daemons include –

- ✓ NameNode
- ✓ DataNode
- ✓ Secondary NameNode
- ✓ JobTracker
- ✓ TaskTracker

❑ Master/slave architecture
- NameNode and JobTracker are masters
- DataNode and TaskTrackers are slaves



Topology of a Typical Hadoop Cluster

Hadoop Distributed File System (HDFS)

- Hadoop employs a **master/slave architecture for both distributed storage and distributed computation**. The distributed storage system is called the **Hadoop File System, or HDFS**.

- The NameNode is the master of HDFS.
  - directs the slave DataNode daemons to perform the low-level I/O tasks.

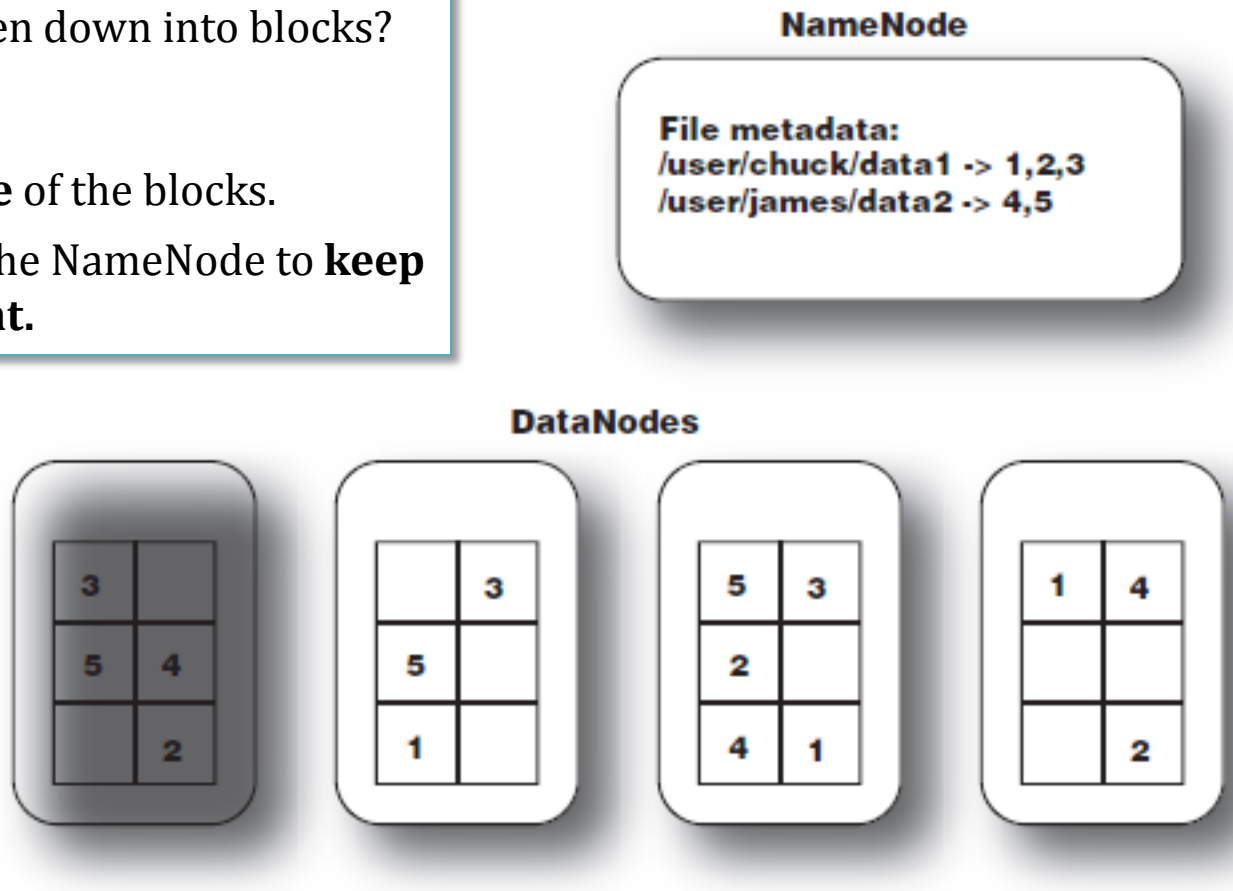
- Each slave machine in your cluster will host a **DataNode** daemon to perform the grunt work of the distributed file system.

  - **reading and writing HDFS blocks** to actual files on the local filesystem.

  - When you want to read or write a HDFS file, the file is broken into blocks and the **NameNode will tell your client which DataNode each block resides in.**

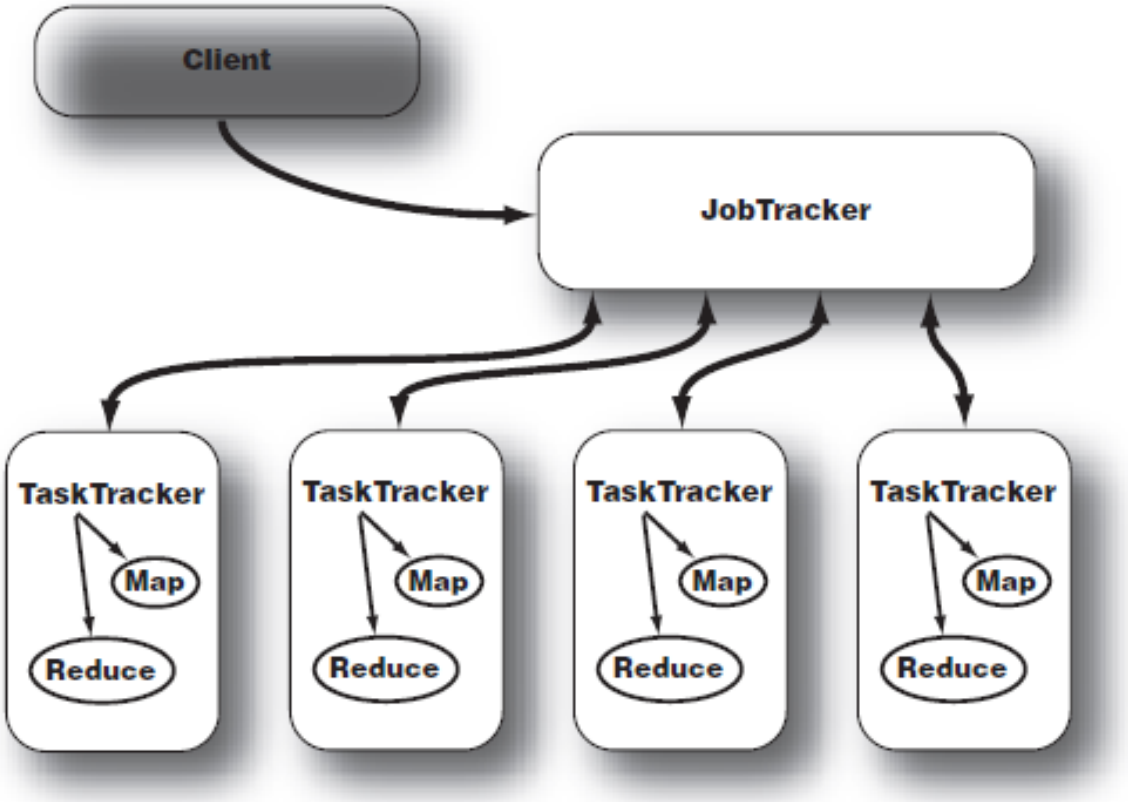
NameNode/DataNode Interaction

- The NameNode **keeps track of the file metadata.**
  - Which files are in the system?
  - How each file is broken down into blocks?
- The DataNodes
  - provide **backup store** of the blocks.
  - constantly report to the NameNode to **keep the metadata current.**



JobTracker/TaskTracker Interaction

- After a client calls the JobTracker to begin a data processing job, JobTracker
  - **partitions the work.**
  - **assigns different map and reduce tasks to each TaskTracker** in the cluster.



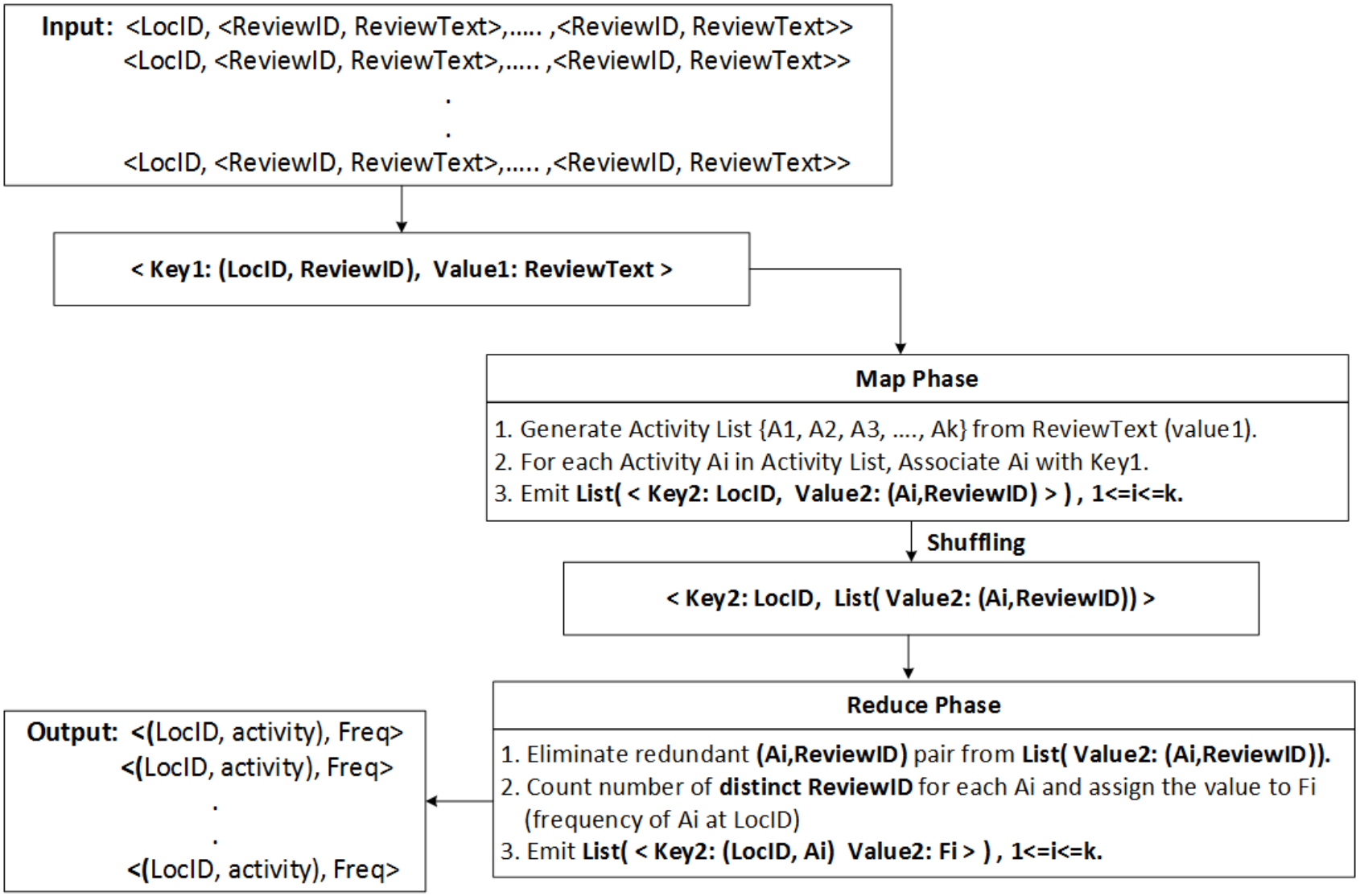
The Problem of Scalability!

What if we want to discover **activities for a large number of locations across globe** and for which we **need to process millions of reviews ?**

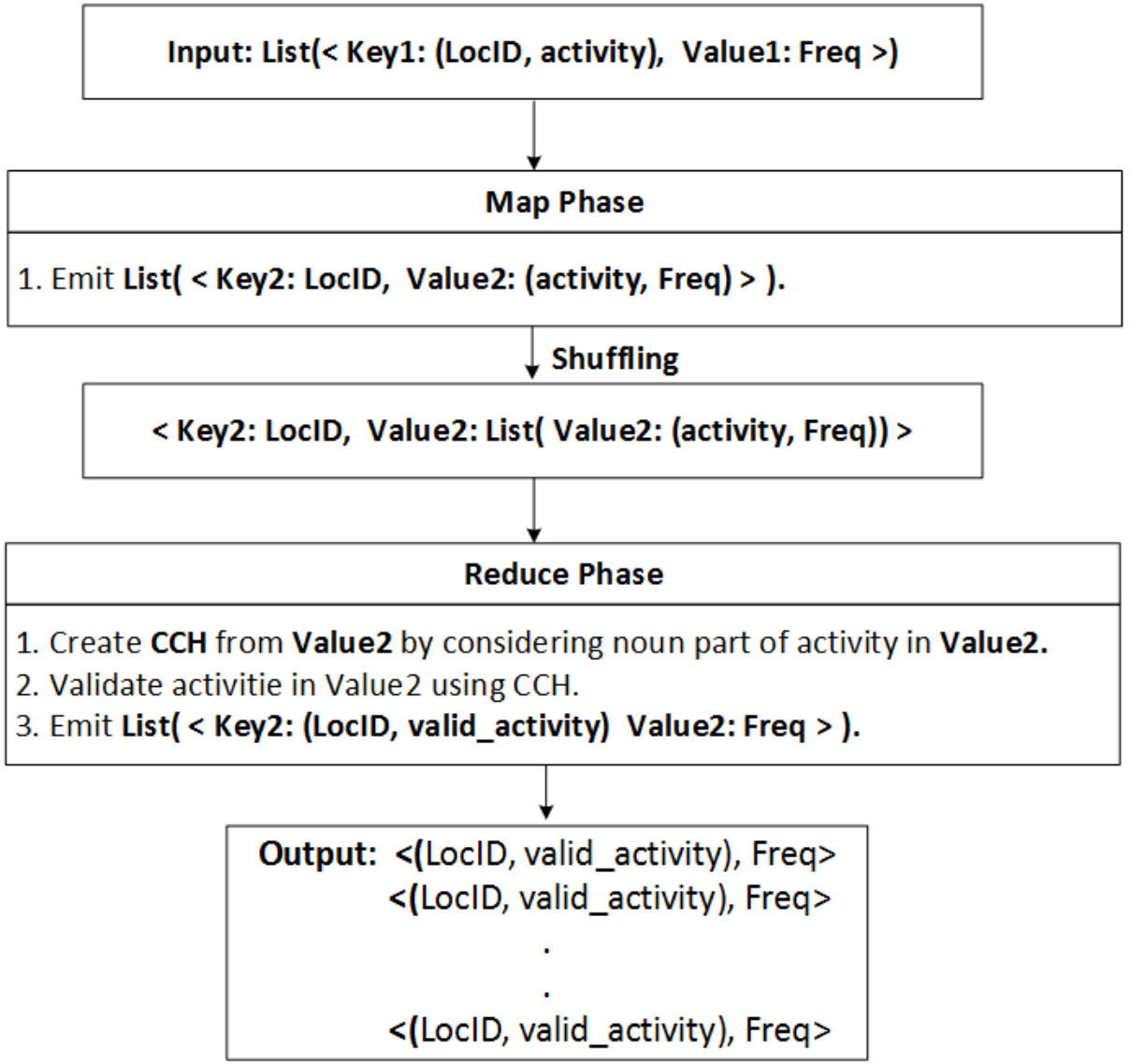
Sequential Execution of **ActMiner** simply cann't solve the problem.

We need to execute "ActMiner" in Distributed Environment
i.e., A Map-Reduce implementation of ActMiner!

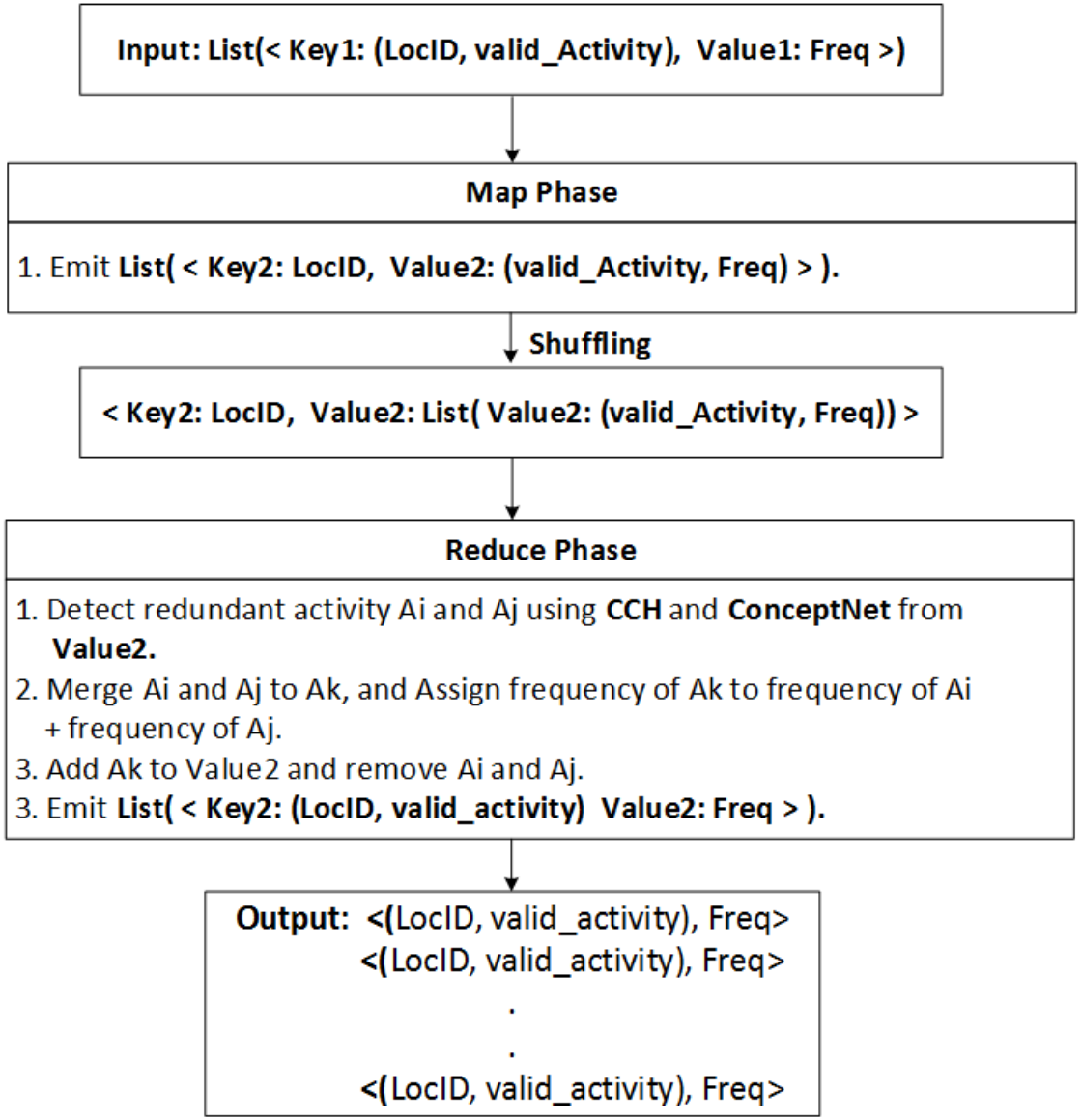
Discover Phase in Map-Reduce Paradigm!



Filter Phase in Map-Reduce Paradigm!



Merge Phase in Map-Reduce Paradigm!



Venue for Future Research Work!

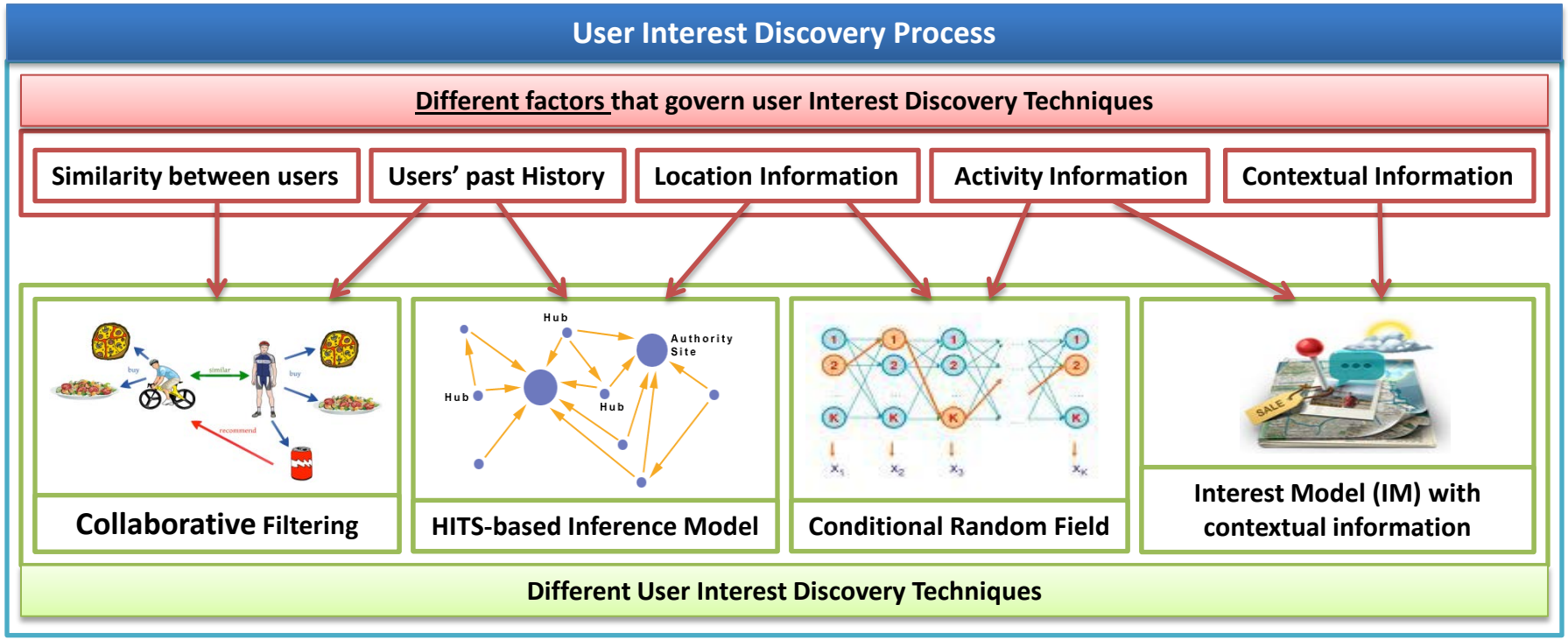
- **An important Observation!**

  "Human behaviour, movements , activities and their frequency of occurrences are driven by his/her Interests".

- Discovering user's interests using location-aware reviews is little less explored.

Let's review Some existing approaches for discovering user Interest

Venue for Future Research Work!

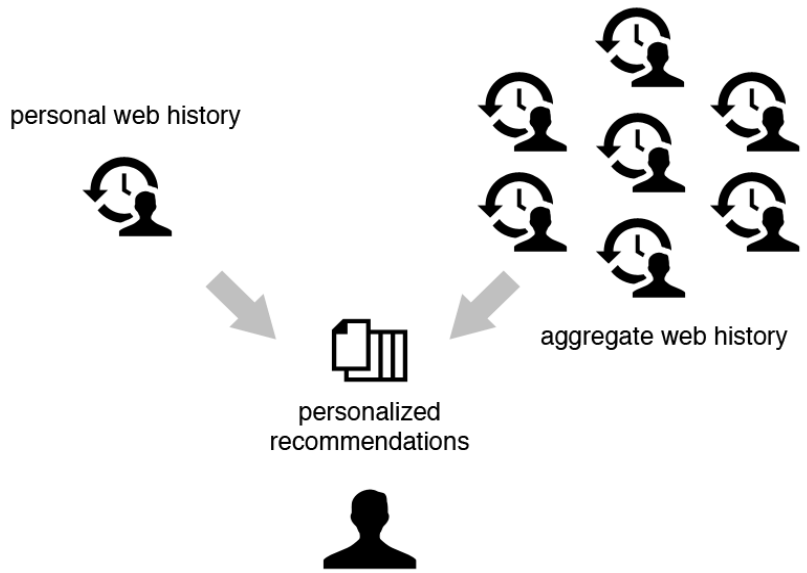


Factors Considered for User Interest Discovery

- **User's Past History.**
- Similarity with Other users.
- User's Location Information.
- User's Activity Information.
- Temporal Information.
- Neighbourhood / Contextual Information.

❖ *By mining users' past history, we can infer about users' behavioral characteristics as well as preference patterns*

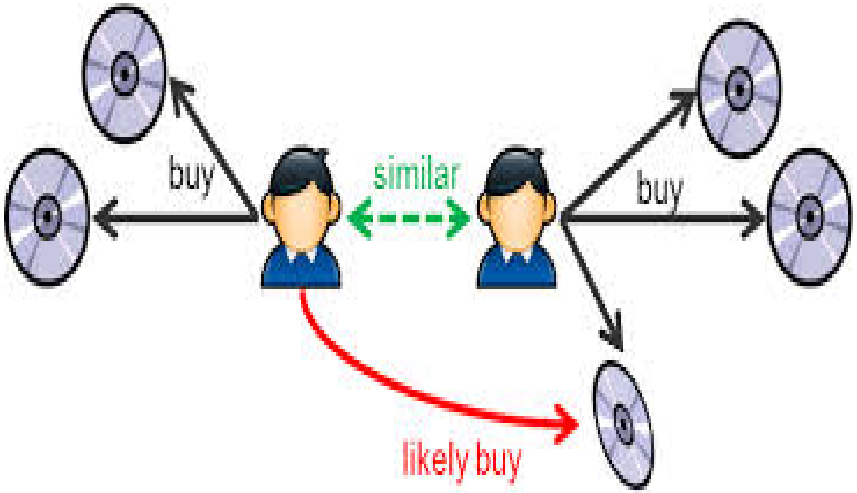
Flipora uses browsing history to make recommendations



Factors Considered for User Interest Discovery

- User's Past History.
- **Similarity with Other users.**
- User's Location Information.
- User's Activity Information.
- Temporal Information.
- Neighbourhood / Contextual Information.

❖ ***Basic Idea: To check the interests of like-minded people with the concerned users and hence, recommend him/her accordingly.***



Factors Considered for User Interest Discovery

- User's Past History.
- Similarity with Other users.
- **User's Location Information.**
- User's Activity Information.
- Temporal Information.
- Neighbourhood / Contextual Information.

- ***Visiting a location is always associated with a purpose or interest.***
- ***By analyzing semantics of the location, Frequency of visits by an user and no. of people visiting that location can help us to discover location based interests.***
- Majority of the existing works have only used GPS information.

Factors Considered for User Interest Discovery

- User's Past History.
- Similarity with Other users.
- User's Location Information.
- **User's Activity Information.**
- Temporal Information.
- Neighbourhood / Contextual Information.

❖ ***Sequence of activities performed by the user at different locations at different times of the day can tell us, what kind of activities the user is interested in.***

❖ *Activities have been mostly discovered using mobile trace information, user annotated history and semantics of the locations they visit.*

Factors Considered for User Interest Discovery

- User's Past History.
- Similarity with Other users.
- User's Location Information.
- **User's Activity Information.**
- Temporal Information.
- Neighbourhood / Contextual Information.

- ***Sequence of activities performed by the user at different locations at different times of the day can tell us, what kind of activities the user is interested in.***
- *Activities have been mostly discovered using mobile trace information, user annotated history and semantics of the locations they visit.*

Factors Considered for User Interest Discovery

- User's Past History.
- Similarity with Other users.
- User's Location Information.
- User's Activity Information.
- **Temporal Information.**
- Neighbourhood / Contextual Information.

❖ *At what time a user visits a location or does an activity and how long he/she stays there or performs that activity is an important factor for the location and activity based interest inference.*

❖ *At what time a user visits a location may be available but how long he/she stays there or performs that activity is not easily available.*

Factors Considered for User Interest Discovery

- User's Past History.
- Similarity with Other users.
- User's Location Information.
- User's Activity Information.
- Temporal Information.
- **Neighbourhood / Contextual Information.**

❖ ***With whom the user spends most of the time and the context of the user based on different times of the days affects user's interests to a great extent !***

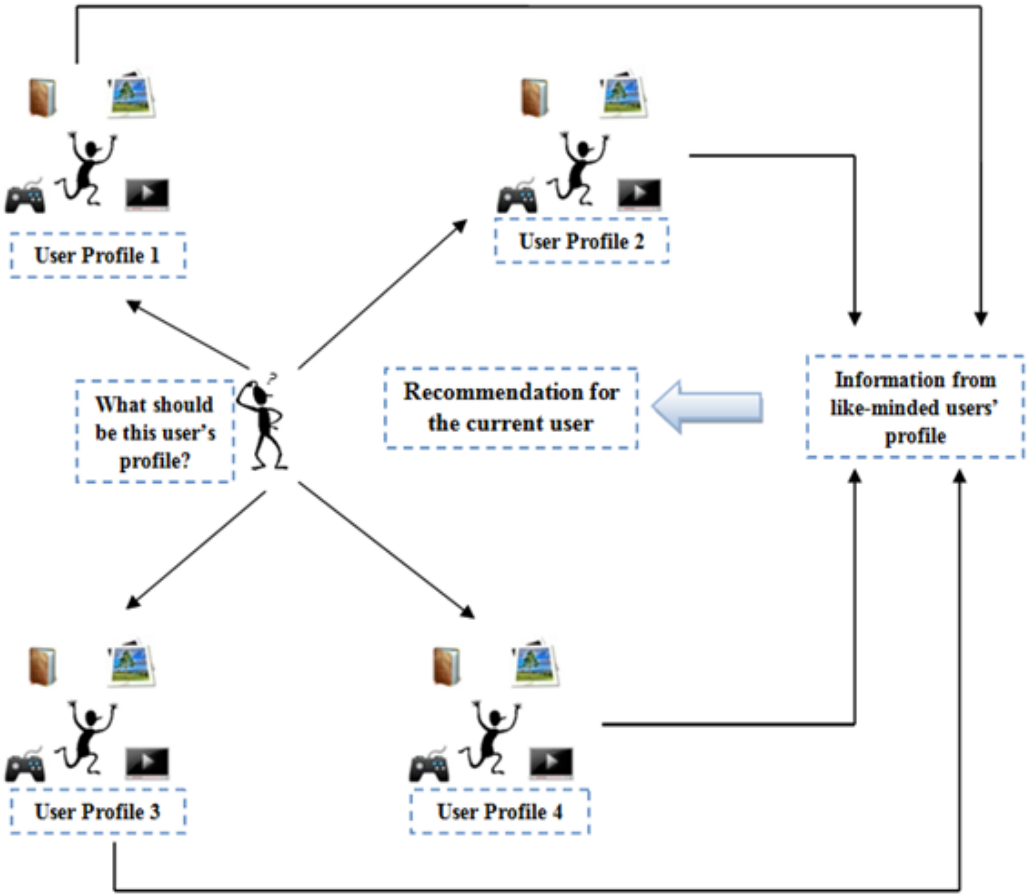
Collaborative Filtering

- Uses the **known preferences of a group of users** to make recommendations or predictions of the unknown preferences for the other users.
- **Fundamental assumption of CF:** if users X and Y rate n items similarly or have similar behaviors, will rate or act on other items in similar way.
- **Three Basic steps :**

✓ Collecting users' opinions via ratings

✓ Searching for like-minded users by comparing ratings and

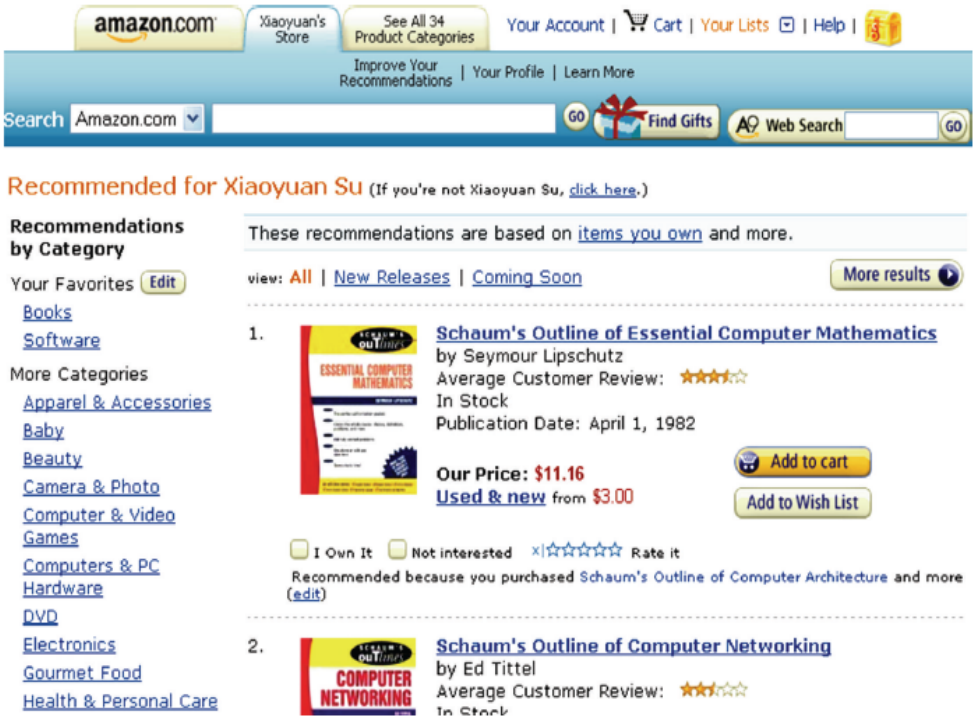
✓ Making recommendations



Collaborative Filtering

- **Some of the challenges faced by CF:**
  - ✓ Dealing with highly sparse data.
  - ✓ To scale with the increasing no. of users and items.
  - ✓ To make satisfactory recommendation in a short-period of time.
  - ✓ To deal with synonymy.
- **Three basic categories of CF:**
  - ✓ Memory-based CF.
  - ✓ Model-based CF.
  - ✓ Hybrid CF Recommenders.
- **A real-world Example:**

  commercial systems such as amazon.com and Barnes and Noble.



Hypertext Induced Topic Search (HITS) – based Inference Model

- A search-query-dependent ranking algorithm for web information retrieval.

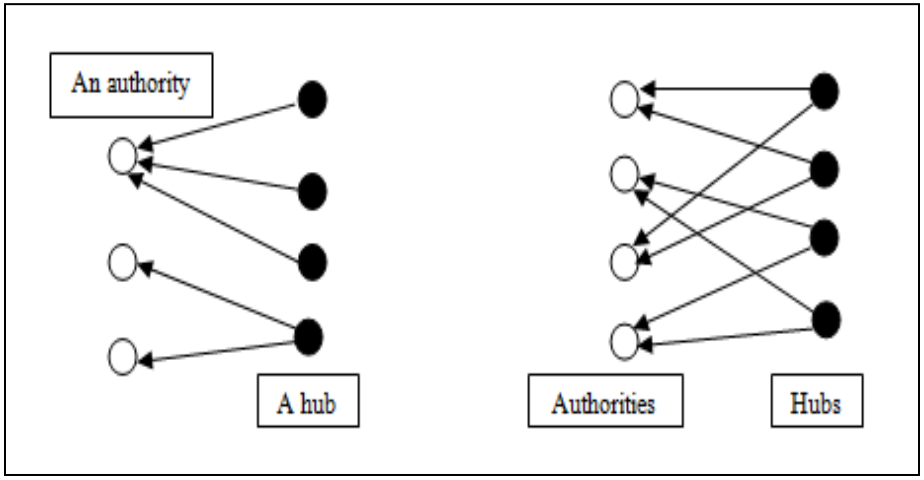
- **Basic Steps:**

✓ Expands the list of relevant pages returned by a search engine.

✓ produces two rankings for the expanded set of pages, authority ranking and hub ranking.

- **Key Idea :** A good hub points to many good authorities and a good authority is pointed to by many good hubs.

  --> **Mutual reinforcement relationship.**



Application of HITS – based Inference Model

- **Research Problem:** Mining interesting locations and classical travel sequences in a given geographical region based on multiple users' GPS trajectories.

- **Strategy:**

✓ first modeled multiple users' location histories with a tree-based hierarchical graph (TBHG).

✓ used HITS-based inference model for location interest and travel sequence mining purpose.

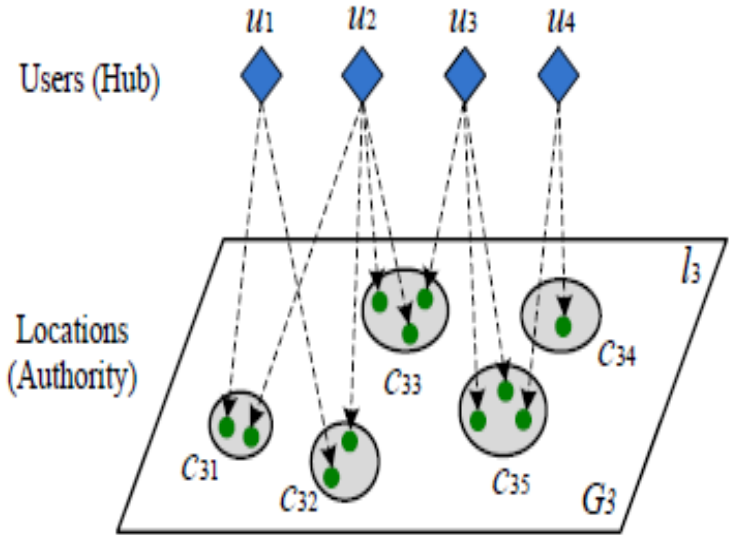
- **Basic Idea of Solution:**

✓ A geospatial region corresponds to a topic;

✓ An individual's hub score stands for their travel experiences,

✓ The authority score of a location represents the interest of the location.

✓ Users' travel experiences and the interest of a place have a mutual reinforcement relationship.



Summary of The Talk …

- We explained **how location-aware review analytics** can **give us an opportunity to build numerous real-life applications** to improve our daily life-style.
- We reviewed **some recent research works** that have leveraged location-aware reviews.
- We also presented **ActMiner** that discovers activities from location-aware reviews and how can we represent the knowledge in the form of a graph database – **LANet**.
- We highlighted **three technical challenges** in location-aware review analytics and **how can we deal with them**.
- Finally, we revealed an **open area of research**: location-aware user interest discovery form location-aware reviews.

Some Useful Resources...

- Apache opennlp developer documentation. source: https://opennlp.apache.org/. Retrieved on December, 2013.
- Conceptnet 5. source: http://conceptnet5.media.mit.edu/. Accessed on December, 2013.
- Neo4j graph database. source: http://www.neo4j.org/. Accessed on December, 2013.
- Stanford log-linear part-of-speech tagger. source: http://nlp.stanford.edu/downloads/tagger.shtml. Retrieved on December, 2013.
- The stanford parser: A statistical parser. source: http://nlp.stanford.edu/software/lex-parser.shtml. Retrieved on December, 2013.
- Wordnet. source: http://wordnet.princeton.edu/. Retrieved on December, 2013.


Thank you.