# FINDING LOCAL EXPERTS FROM YELP DATASET

BY

TANVI JINDAL

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Adviser:

Professor Jiawei Han

# Abstract

Local experts are people who have special expertise in an area, but that expertise is limited to a geographical region. It makes sense to say that someone is a global expert on quantum physics, but it is hard to find someone who knows all about the best ice-cream places anywhere in the world. This is the reason why for local topics, local experts are a better source of information than just experts; they can be a great way to give out a digital word-of-mouth. In this work, we have proposed a system to find local experts on different things from Yelp data. Review and recommendation systems have become a big part of how people consume different products and businesses today. Yelp is a great example of a huge database of reviews on businesses ranging from restaurants, gas stations, salons, and even doctors. The way people consume this extensive database is very much limited to looking at the star rating of the business, which ignores so many other perspectives. We combine various signals from the reviews and spatial data to come up with an algorithm to find the local experts. Yelp provides a large subset of its data for experimentation and we use this dataset to test our hypotheses. Finding local experts can be used in many ways, such as generating weighted, more accurate reviews for businesses, and to create recommendations for new users. If you visit Paris for the first time, you would now be able to get suggestions for food and things to do from native French people who are local experts in those things. The aim of this paper is to automatically find these people who can give you the best local juice on what you want to know.

*To my parents and my friends, who have supported me through my journey*

# Acknowledgment

I would like to take this opportunity to express my deep gratitude to Professor Jiawei Han, my research advisor for his patience, guidance, and support through every step of my graduate school. Other than being an amazingly helpful and patient advisor overall, he was also always available to discuss the intricacies of the work and provided valuable suggestions and critiques to help me move forward. I would also like to thank all the students in my research group, specifically Chao Zhang, who had regular discussions with me and gave very helpful suggestions and insights and was an integral part of my thesis project. Finally, I wish to thank my parents and my friends for giving me support and encouragement throughout my study.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The aim of this work is to come up with a system that can find local experts given a business category. Local experts differ from general topic experts in the sense that their expertise is very limited to a geographical location. Local experts can play a very important role in addressing local information queries, like ”where can I find the best Indian street food in the San Francisco bay area?”, ”which cafes have performances from local singers?”, ”who is a good dermatologist in the area?”. There have been studies that show that people are likely to prefer learning from local experts who know the neighborhood well and have first hand experience [2].

Especially in the context of Yelp data and in the quest of discovering the best local businesses, local experts are the best possible source of information. If you are in the Bay area and you feel like eating spicy Indian food, you would rather take advice from an aficionado living here, rather than one in New Delhi, even though the latter might be more knowledgeable in Indian street food as a whole. If you have ready access to suggestions and advice from local experts, it can be like visiting Paris for the first time, and have a local show you around the cool, local places you would never find on a ”Things to do in Paris” list.

A lot of new businesses have also come up that revolve around the idea that people really like to see new places and get new experiences from the eyes of a local. There are many startups, like Vayable [15] that sets you up with a local connoisseur when you visit a new place, so you can have a very different and authentic experience. One of the billion-dollar startups of recent times, AirBnb [1] is based on the idea that travelers like to experience the lifestyle of local people up,

close and personal by living as a guest in their house rather than staying in a hotel. There are many food-based startups which source delicious recipes from local chefs and deliver food to your doorstep.

## 1.1 Related Work

Finding local experts is not a trivial task, and the way people discover local experts has changed over time. Traditionally, people have relied on word-of-mouth, or manually curated lists published in magazines, or more recently in blogs, like "10 best broadway shows to catch when you visit New York city". Over time, a lot of online platforms have come up for sharing information and expertise. There are forums and question-answering sites, for example, Yahoo! answers, StackOverflow for programming related questions, StackTex for specifically LaTEX related queries. There is a huge community of people who invest their knowledge to help other people on these communities.

### 1.1.1 Finding experts

Web-based communities and forums have become an important place where people go to seek advice and expertise on various topics. The topics range from household things like "how to get wine stains out of my carpet" to programming advice like "how to release a unique pointer in C++". Jun Zhang et al [18] used various network-based signals and algorithms to automatically find experts from such question-answering systems. Liu et al [12] used signals from the users' profiles and the type of questions he/she answered in the past for expert discovery. There have been other attempts for finding experts in enterprise corpora too [3, 5] using internal documents and email communications. These approaches are however focused on finding general topic experts rather than local experts.

Other than question-answering communities, people seek advice on social networks also. Face-

book and Twitter are an online manifestation of the word-of-mouth spread of information, as they have no structured review system. Social networks and recommender systems have changed the way people discover, explore and evaluate new products, businesses, and services. Recommender systems have become an integral part of businesses like Amazon, Netflix, and so on. Other than users being able to rate and review products, there is also a social aspect to recommendation systems where people can take inspirations from their friends' opinions and factor them into making their purchase decisions. Recommender systems are generally content-based, which depend on the user's history; or collaboration-based, which depend on the social network of the user; and most effectively, hybrid systems, which use a combination of both to make more informed recommendations.

The widespread availability of affordable GPS-enabled connected devices has made it possible for users to very easily share their location. Social networks like Facebook, Twitter, Foursquare have users' location attached to most of the posts, which makes it possible to create systems that can use both location and domain expertise to find local experts. Antin et al [2] conducted a study about people's attitudes towards their local knowledge and personal investments in their neighborhoods. They found that a good majority of people consider themselves to be local experts and are also ready to be contacted by others for advice about local topics.

There have been a couple of works where people use Twitter data to find local experts. Li et al [10] proposed a framework for describing a user's geo-spatial domain expertise in microblog settings. They used a point-of-interest based topology to understand users' activities related to a place, in order to design a classification system that can predict the local expertise of a user. Cheng et al [6] worked on a system to identify local experts in various categories using Twitter lists, and how users interact with them. Their system is the closest to the one proposed in this work, as they are trying to solve the very same problem. They propose LocalRank, which is a two-pronged effort at identifying experts. It combines *topical authority*, which tells how knowledgeable the

user is about the topic, and *local authority*, which tells how closely is the user associated with this location. They make use of the information-rich twitter lists, where each user also has a location, and the linkages between users through these lists, to estimate local authority. For topical authority, they adapt a language modeling algorithm and augment it to incorporate the distance-weighted social ties of users.

### 1.1.2 Analyzing Yelp data

Yelp has made it possible for people to rate and review local physical businesses just like they would do for products online. The overall rating for a business is a simple average of all the ratings for that business. While exploring a business, users generally just make their decision based on this average star rating. Although it does give a pretty good idea about what people think about the business, but it ignores so many other pieces of information, including the attributes of the business, the users, and the review text itself.

A lot of work has been done to consume this information in much better ways. Yelp has been organizing dataset challenges for a few years where students use Yelp data to come up with fun, innovate and creative ways to generate great new insights and opportunities. Over the years, there have been very good papers that have come out of these challenges. I will talk about some of these papers and the different ways in which Yelp data can be consumed for interesting analyses. There are different categories of these papers, some focus on the Yelp social network, some focus on getting utility out of the review text by topic analyses, and some other try to discover trends.

Felix W. [16] focussed on understanding the utility of social understanding systems. He studies how recommendations diffuse through the network and proposes an algorithm for social networks that will optimize the utility of an individual by making connections leading to good recommendations, and the utility of the network in effective information propagation at the same time. Another very interesting paper by Hood et al [4] looks at a very comprehensive set of features to predict

a star rating for a business. The features involve time-dependent features including number of reviews, number of days since last review; text-based features including sentiment analysis of the keywords in each category; and user-clustering features, including number of reviews the user has written.

Among the papers that focus on understanding the review text by doing topic analysis, McAuley et al. [13] find hidden factors and topics in the text to justify the rating accompanying the review. James Huang et al [7] discovered latent subtopics from review texts using online LDA, and predicted star ratings for each of the latent topics. In simple terms, they describe that a 3 star rating for a restaurant might actually be a result of combination of 4 star rating for food, and a 2 star rating for the ambience and service. In a very similar spirit, Jack Linshi [11] modified LDA to codeword-LDA which looks at hidden subtopics in the text, but uses composite subtopics, by looking at a positive or negative adjectives associated with the terms, creating topics like ”good food”, ”bad service”, ”bad bartender”, instead of simple ”food”, ”bartender”, ”good”, which gives a much better picture of what the review is saying, and then uses these to create personalized ratings for user. A user might be more interested in good food without caring about the ambience, and hence, should see a different rating for the same restaurant than someone who cares a lot about the ambience.

## 1.2 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 talks about the Yelp dataset in detail and recognizes simple trends and characteristics of the data. It helps to understand the intricacies of the data so that we can use it in the best possible way. Chapter 3 describes the problem of finding local experts formally and then introduces the overall algorithm devised to find local experts from Yelp data. It also talks about the two major parts of the algorithm: understanding topical authority

and local authority, and how to combine these two to get the local experts on a given topic. It also talks about feature extraction and the learning process. Chapter 4 then talks about the experimental setup and details the results obtained from a couple of different topics. Chapter 5 is a conclusion discussion and also outlines the possibilities of future work in this area.

# Chapter 2

# The Dataset

Yelp is an online database of local businesses and provides a great way for users to explore, rate and review the businesses they visit. Businesses create and maintain their own listings, and try to highlight the products and services that will appeal to users and attract them to visit and rate the business. Yelp covers a vast variety of businesses, like restaurants, bars, cafes, local events, doctors, pharmacies, hotels and so on. Users have accounts, and can also add friends on Yelp. Yelp maintains its own social network with a graph of who is who's friend, rather than depending on an external social network like Facebook, or LinkedIn, unlike a lot of other review and recommendation systems. Users can give a star rating from 1 to 5 for a business, and can also write a text review justifying the number rating. These ratings serve as a great metric for users who are exploring local business, and help them in judging which one would be the best for them, making Yelp a great recommendation system. Each business has an overall rating which is just an average of the star ratings for all the reviews that the business has reviewed. Users can also vote for reviews written by other users, the votes can be given as funny, cool, or useful.

Over the years, Yelp has collected a huge database of users, businesses, and reviews. Yelp has made a subset of the data available to the scientific and academic community by organizing challenges, and has opened up a lot of opportunities for analyzing and using this data for different purposes in different ways. It is a huge dataset of 1.6M reviews and 500K tips by 366K users for 61K businesses with 481K business attributes, e.g., hours, parking availability, ambience. This dataset also incorporates a social network of 366K users for a total of 2.9M social edges. The dataset includes businesses in four different countries : Edinburgh, U.K.; Karlsruhe, Germany;

Montreal and Waterloo, Canada; Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison, U.S., making it a very versatile dataset.

The dataset has different data types: businesses, reviews, users, checkins and tips. Following is a description of all the fields in each of the data types.

## 2.1 Business

```
{
    "type": "business",
    "business_id": (encrypted business id),
    "name": (business name),
    "neighborhoods": [(neighborhood names)],
    "full_address": (localized address),
    "city": (city),
    "state": (state),
    "latitude": latitude,
    "longitude": longitude,
    "stars": (star rating, rounded to half-stars),
    "review_count": review count,
    "categories": [(localized category names)]
    "open": True / False (corresponds to closed, not business hours),
    "hours": {
        (day_of_week): {
            "open": (HH:MM),
            "close": (HH:MM)
```

```
        },
        ...
    },
    "attributes": {
        (attribute_name): (attribute_value),
        ...
    },
}
```

Each business has a business id, location information including latitudes and longitudes, full address, and so on. It also has the number of reviews that have ever been written for the business, and an average star rating across all the reviews. Yelp also stores other information like hours, parking situation, ambience, and so on.

## 2.2 Review

```
{
    "type": "review",
    "business_id": (encrypted business id),
    "user_id": (encrypted user id),
    "stars": (star rating, rounded to half-stars),
    "text": (review text),
    "date": (date, formatted like "2012-03-14"),
    "votes": {(vote type): (count)},
}
```

Each review consists of a star rating and an optional review text, possibly justifying the star rating. The review datatype also connects the users to the businesses they review. Each review can also

get votes from other users, who can say if they find the review cool, funny, or useful. Figure 2.1 gives a distribution of the number of reviews written by users. To make the graph more readable, it excludes the long tail with number of users less than 10, and also the points where number of reviews is less than 30.

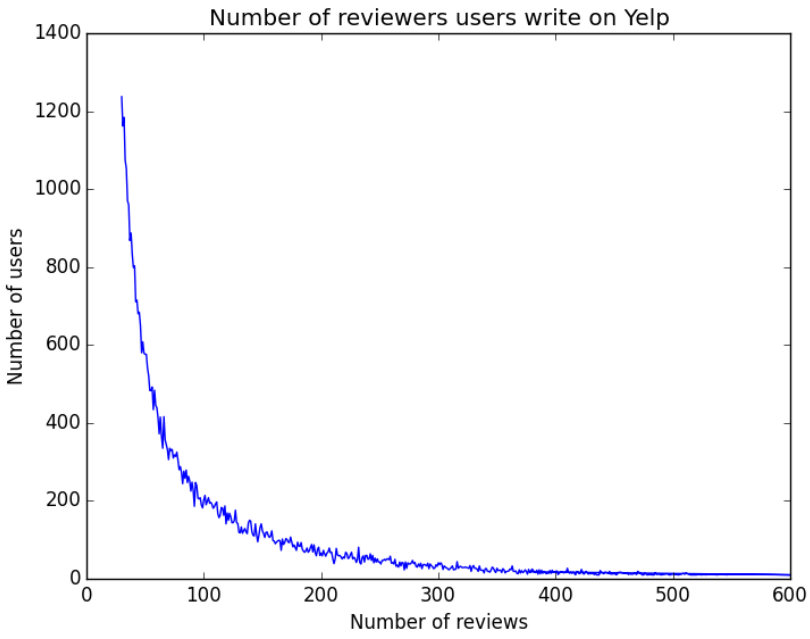



Figure 2.1: Distribution of number of reviews

## 2.3 User

```
{
    "type": "user",
    "user_id": (encrypted user id),
    "name": (first name),
    "review_count": (review count),
    "average_stars": (floating point average, like" 4.31),
    "votes": {(vote type): (count)},
```

```
    "friends": [(friend user_ids)],
    "elite": [(years_elite)],
    "yelping_since": (date, formatted like "2012-03"),
    "compliments": {
        (compliment_type): (num_compliments_of_this_type),
        ...
    },
    "fans": (num_fans),
}
```

Yelp has a rich user network, where it stores not just the basic information about them like their name, the number of reviews they have written, how long they have been using Yelp for; but it also creates its own social network by having friend relations between the users. Users can even be fans of each other.

To get a better picture of how strong the Yelp social network is, I did an analysis to see how many friends users generally have. Not very surprisingly, a large number of users have no friends, and a lot of them have less than 5 friends. But at the other end of the spectrum, there are users with a huge number of friends, around 75 users have more than 1000 friends! Figure 2.2 is a plot of the number of friends with the number of users who have that many friends. To make the graph more readable, it excludes the long tail with number of users less than 10, and also the points where number of friends is less than 20.

## 2.4 Checkin

```
{
    "type": "checkin",
    "business_id": (encrypted business id),
```

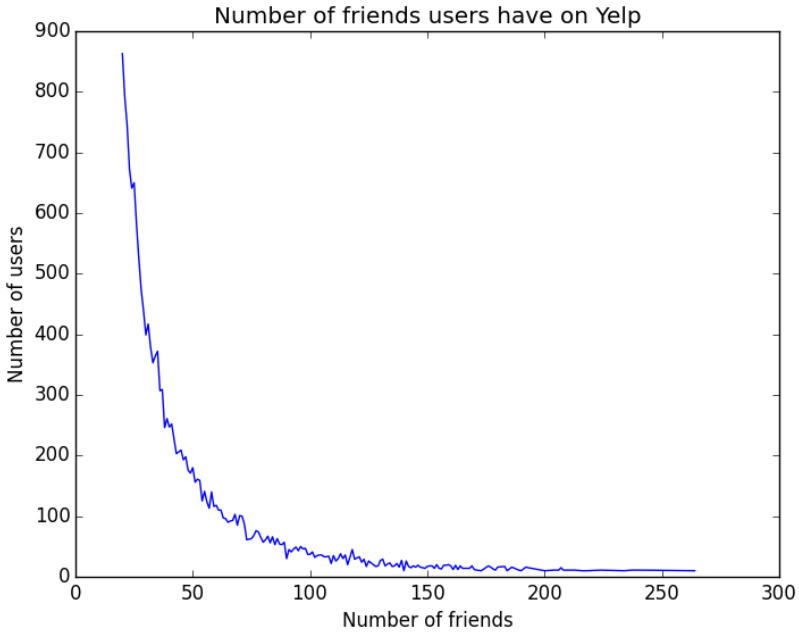



Figure 2.2: Distribution of number of friends

```
    "checkin_info": { (no. of checkins in time periods)
        "0-0": (from 00:00 to 01:00 on all Sundays),
        "1-0": (from 01:00 to 02:00 on all Sundays),
        ...
        "14-4": (from 14:00 to 15:00 on all Thursdays),
        ...
        "23-6": (from 23:00 to 00:00 on all Saturdays)
    }, # if there was no checkin for a hour-day block
       # it will not be in the dict
}
```

This gives an aggregated view of all the checkins for a business for every hour of the day, for every day of the week, and gives a great idea about what are the busiest times for the business.

## 2.5 Tip

```
{
    "type": "tip",
    "text": (tip text),
    "business_id": (encrypted business id),
    "user_id": (encrypted user id),
    "date": (date, formatted like "2012-03-14"),
    "likes": (count),
}
```

Tips stores random comments that users leave about a business, they are different from reviews in that they don't have a star rating, and are just quick indications for others.

# Chapter 3

# Finding Local Experts: Algorithm Design

A local expert is someone whose expertise in a particular subject is limited to one particular geographical area. The aim of this work is to find local experts in a given category, for example, to find the local expert in Irish bars. The following definition puts this problem in a formal manner.

## 3.1 Problem Description

**Definition 1** (Local expert finding)**.** *Given a query q to find local experts in a category c(q) in a location l(q), find a set of users that are knowledgeable in category c(q) and are local to location l(q).*

Given the entire set of users U = $\{u_1, u_2, u_3, ..., u_i, ...u_{n-1}, u_n\}$, assuming that n is the total number of users in the Yelp dataset. Each user $u_i$ can be associated with a location $l_i$ and a set of categories $C_i$ which the user has knowledge about. The objective is to find the set of users U(q) such that for each $u_j$ in U(q), c(q) is in $C_j$ and l(q) is close enough to $l_j$.

To understand whether the two locations, the user's location, and the query location, are close enough to each other, we devise a way to find the distance between the two locations and then use a distance threshold $t_d$ on this distance. If the distance is less than the threshold, we say that the user is local to the query location.

## 3.2 Overall Algorithm

There are two components to establish a user as a local expert:

(a) *Topical authority*: This part understands whether the user has considerable knowledge about the category to be considered an expert or not. In this work, we train a classifier given the category, which puts users into two bins: experts and non-experts.

(b) *Local authority*: This part establishes whether the user is local to the area, basically understanding if she lives in the locality and visits and reviews a lot of businesses in that category. In this work, we use a threshold, $t_d$ on the distance between user location and query location to decide whether the user has local authority or not.

These two components are combined together in a very simple overall algorithm. We start by training a topic expertise classifier for the query category c(q). The we find a set of possible experts by passing each user through the classifier. Finally from this set of potential experts, we keep only the ones that are close enough to the query location. The following is the algorithm:

```
candidate_experts = []  # list of potential local experts
local_experts = []  # list of final local experts
for each user u:
        if she is an expert on the category:
                Add u to candidate_experts
for each candidate user v in candidate_experts:
        if distance(queryLocation, userLocation) < distanceThreshold:
                Add v to local_experts
```

The following sections describe how we estimate the user's location, and how we then calculate the distance between user location and query location. And then we talk about topical authority and the feature set that we use to train the classifiers for topical authority.

## 3.3 User Location

For the Yelp dataset, the users do not have an associated location, and so there is no direct way to localize the user. So we came up with a way to do so using the businesses that the user visited and reviewed. The idea is that a user would visit the most businesses in the locality she lives in, other than maybe travels or permanent moves, so we can get a model of her location using a distribution of the locations of businesses she has reviewed. From the review dataset, we can obtain a list of business ids of the businesses that the user has ever visited and then using the business dataset, we can convert these business ids to locations. We wrote a simple map reduce script to effectively get a set of locations for each user.

The map reduce program has two steps: the first one with a mapper (Mapper1) and a reducer (Reducer1); and the second one with just a reducer (Reducer2). Both business and review data is fed into Mapper1, and it emits two types of key-value pairs, based on the type of data. For businesses, it emits business id (bid) as key and (location: $[latitude, longitude]$), where $[latitude, longitude]$ is the location of that business, as value. For reviews, it again emits business ids as keys and (user : user id) , with the user id (uid) for the user who wrote the review as values. So each Reducer1 gets the location of a business and all users who have ever written reviews for it, and emits (uid, $[latitude, longitude]$) pairs. Reducer2 then appends all locations for each user and emits (uid, {list of locations}, hence giving us a list of locations for each user.

Given the set of locations for each user, we tried to visualize them by plotting them on a map. For a general user, you would expect her to mostly visit businesses in and around the neighborhood/city she lives in. There can be outliers in case she travelled and then wrote a review for a business there. Also, the neighborhood of a user changes when she moves from one city to another. For example, if Julie is an active Yelp reviewer and she went to school in Los Angeles and then moved to New York for a job, she would have a good number of reviews for both LA and NY

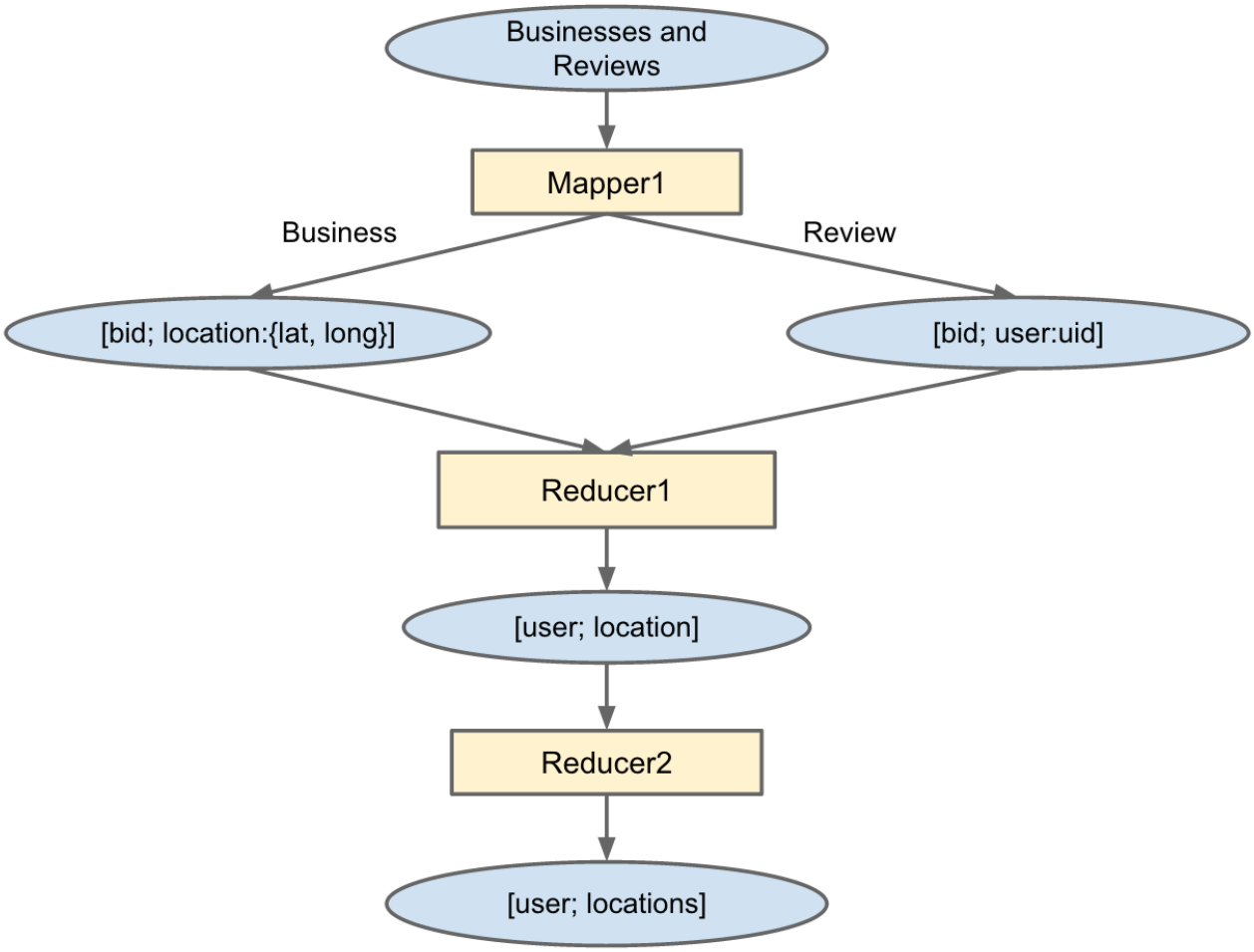
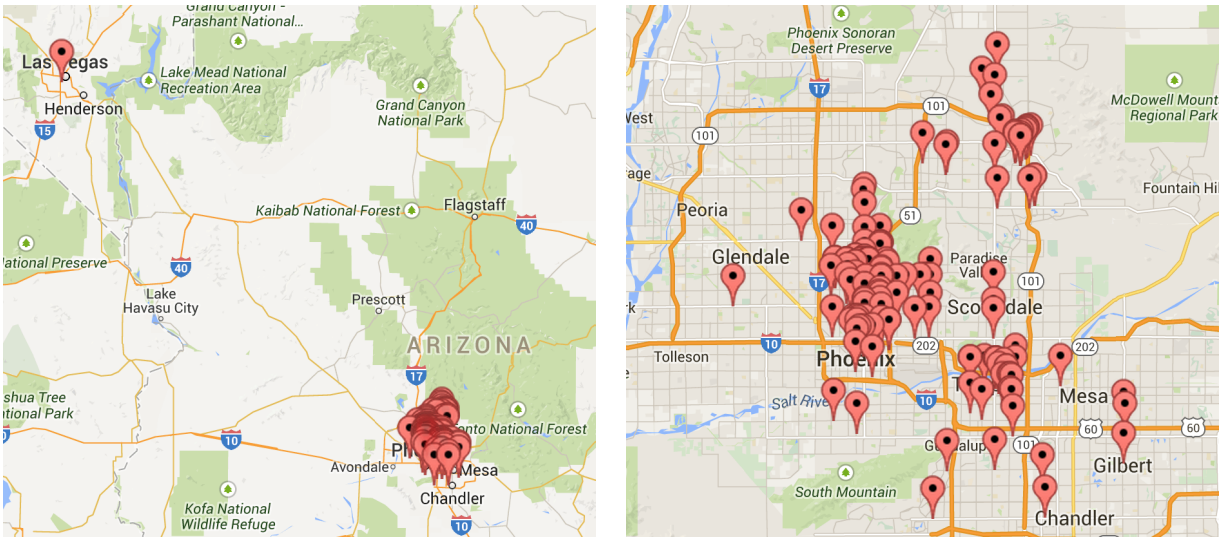



Figure 3.1: Map Reduce for user locations

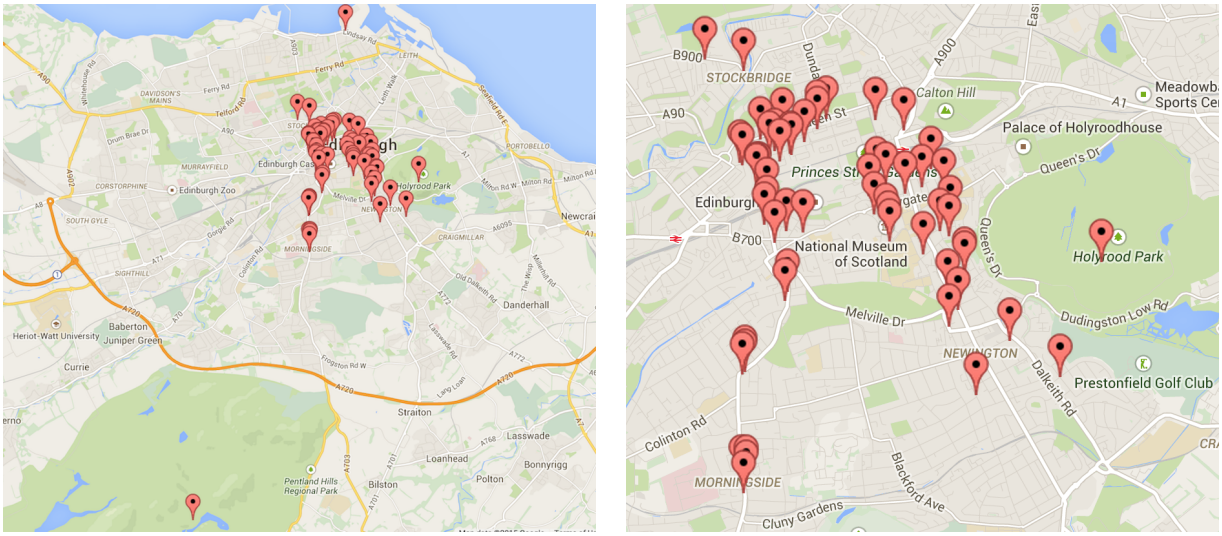
businesses. We normally expect a small number of such localities for a user. This was verified by plotting the locations for some of the most active users on a map. We can easily see how most of the users have locations that are centered around one or a few locations and spreading out from there. Figures 3.1, 3.2, and 3.3 plot the locations for three active Yelp users, where users u1 and u2 have just one major center, but user u3 has 2 main centers.

There are many possible ways in which we can determine the location of a user given the set of locations she visited. There have been many attempts to use location history to characterize users




(a) All points (b) Zoomed in

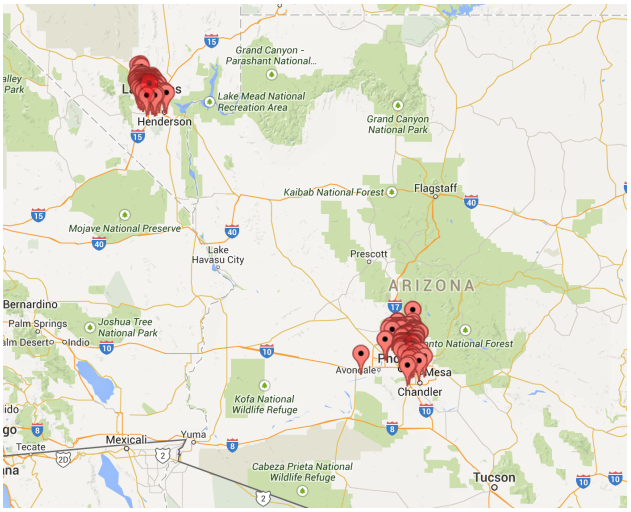
Figure 3.2: Location points for user1



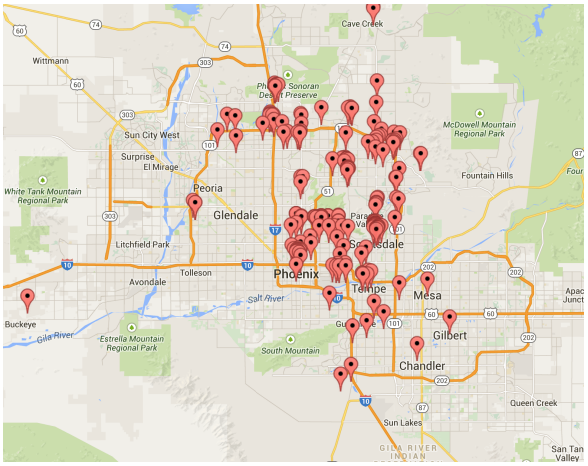

(a) All points (b) Zoomed in

Figure 3.3: Location points for user2

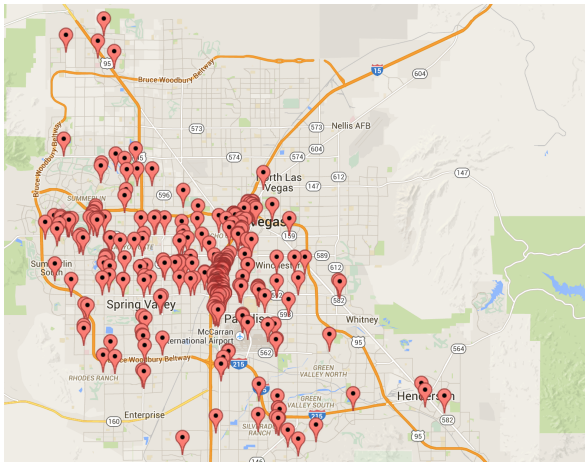
[9, 17, 19], but in this case, we simply want to model this data to get to a location estimate for the user. There could be very simple ways like taking the geographical centre of all the locations, but that would get highly skewed by outliers (think of long-distance, foreign travels). We could think of excluding outliers and then taking the centre of the remaining locations, but again that is not the best way to model the location, and we decided to use a gaussian mixture model.




(a) All points, showing two clusters




(b) Zoomed in on Pheonix cluster




(c) Zoomed in on Las Vegas cluster

Figure 3.4: Location points for user3

### 3.3.1 Guassian mixture model for locations

Given the location data distribution, Gaussian Mixture Model seems an intuitive choice to effectively model the location of users. Gaussian Mixture Models (GMMs) is a sum of Gaussians where each has its own mean and covariance. In our case, it models users locations as centered around multiple locations and a little spread around each center. The model consists of the means, covariances, and a probabilistic assignment of every data point to the Gaussians. Given a dataset, we can fit it to a GMM using Expectation Maximization (EM) by starting with randomly chosen centers and alternating between E and M steps using logarithms and log-sum-exp formula

The notations:

M dimensions

k = 1 ...K Gaussians

n = 1 ...N data points

P(k) population fraction in k

P($x_n$) model probability at $x_n$

$\mu_k$ (The k means, each a vector of length M)

$\Sigma_k$ (The K covariance matrices, each of size MxM)

P($k \mid n) \equiv p_{nk}$ (The K probabilities for each of the N data points)

Overall likelihood of the model, $L = \prod_n P(x_n)$

Specifying the model as a sum of Gaussians, $P(x_n) = \sum_k N(x_n \mid \mu_k, \Sigma_k) P(k)$

And, $N(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{M/2} det(\Sigma)^{1/2}} exp[-\frac{1}{2}(x-\mu) \cdot \Sigma^{-1} \cdot (x-\mu)]$

In our case, with location data, M=2, with the dimensions being the latitude and the longitude. We can then apply the EM algorithm to fit the data to the appropriate GMM.

- E step: If we knew the Gaussians, we could assign the points by relative probability density of each Gaussian at each point.

$$p_{nk} \equiv P(k \mid n) = \frac{N(x_n|\mu_k, \Sigma_k) P(k)}{P(x_n)}$$

- M step: If we knew the assignment, we could estimate the Gaussians by weighted means of the points assigned to each of them.

$$\hat{\mu_k} = \sum_n p_{nk} x_n / \sum_n p_{nk}$$
$$\hat{\sum_k} = \sum_n p_{nk} (x_n - \hat{\mu_k}) \otimes (x_n - \hat{\mu_k}) / \sum_n p_{nk}$$

$$\hat{P}(k) = \frac{1}{N} \sum_{n} p_{nk}$$

The EM algorithm performs multiple iterations of alternating E and M steps. With each iteration, the overall likelihood increases, and eventually the steps converge to maximize the overall likelihood. Figure 3.5 shows the centers found using GMM for user3, whose location distribution is shown in Fig 3.4.


Figure 3.5: Location centers for user3 derived using GMM

## 3.4 Location Authority

Now that we have a model for the user's location, we need to find a good way to measure a sense of localness of the user to the query location. To do so, we assume that the query location, $l(q)$ can be represented in the form of a single latitude-longitude position $p_q$ and an area around it with a radius $r_q$. For the user location, l(u), we have a GMM with means which are basically the centers around which the user's locations are centered. Let the centers for the user location be C(l(u)).

We can define the distance between l(u) and l(q) as follows:

$$d(l(u), l(q)) = min_{c \in C(l(u))}[d(p_q, c)]$$

where $d(p_q, c)$ is the geographical distance distance two lat-long positions between c and $p_q$, using Haversine formula. Finally, we check if $d(l(u), l(q))$ is less than the distance threshold $t_d$. Figure 3.6 shows the distances of a query location from the center location points for user 3. I chose the query location to be my current location in Mountain View. As can be clearly seen from the map,

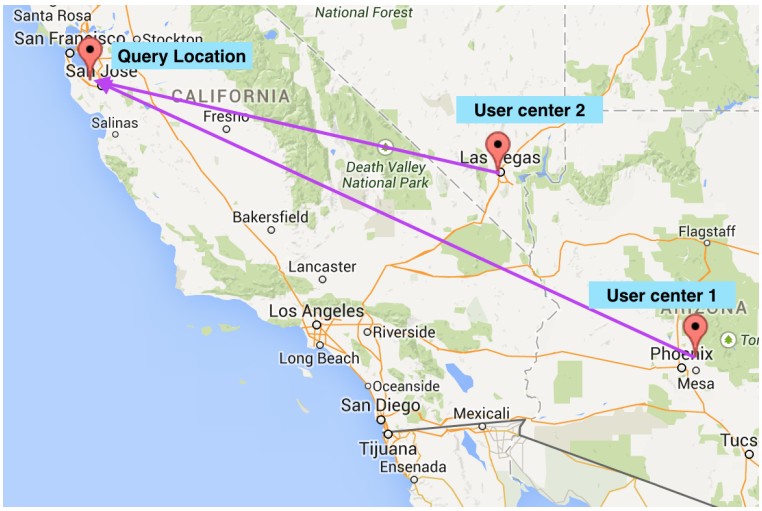



Figure 3.6: Distance of user3 from a query in Mountain View

the Las Vegas center is the one closer to the the query. This means that if someone in Mountain View queries for the local experts in Italian food, and user3 was one of the candidate experts, then the distance between query location and the Las Vegas center would be used to determine if the user has local authority too.

## 3.5 Topical Authority

Yelp organizes all its businesses into categories, where each business can have multiple categories. For example, a chinese restaurant has both ”Restaurants” and ”Chinese” in its categories, and an orthodontist may have both ”Health & Medical” and ”Orthodontist” in its categories. A quick analysis showed that there are more than 700 different categories in the Yelp academic dataset. The topic of the local expert query category c(q) is one of these categories already identified by Yelp.

To understand whether a user has considerable knowledge about a category, we identified a set of features that we can use to train our classifiers with. There can be many different types of features and they are all described in some detail below.

### 3.5.1 User-specific features

These are the features that just depend only on the user, and the way she uses and interacts with Yelp.

- **Yelp age**: This is the time that the user has been active on Yelp, and for our purpose, we measure this as a number of months, using the ”yelping_since” attribute in the user data.
- **Total reviews**: The total number of reviews that the user has ever written for any business. It is a clear indicator of how active the user is on Yelp.
- **Average review length**: This is the average length of all the reviews written by the user, and is an indicator of the quality of the user's reviews. We can use it either as an absolute number or a boolean indicating if the reviews are long enough using a threshold on the review length.
- **Average rating**: This is the average star rating given by the user to businesses.
- **Variance in rating**: Along with the average, the variance gives a better picture of the user's rating habits.
- **Number of friends**: The number of friends that a user has on Yelp is a good indicator of how many people get influenced by her ratings and reviews, and also how many users influence her.

### 3.5.2 User category features

These are the set of features that describe how the user behaves given a category C. Some of these are similar to the user-specific features, but describe similar numbers for this category category.

- **Total reviews in C**: This is the total number of reviews that the user has written for any business that belongs to category C. A user may have written hundreds of reviews overall but if we are trying to find local experts in Mexican food, what matters more is the number of reviews for Mexican restaurants. For e.g., user A with 10 total reviews and 6 about Mexican restaurants is a better candidate than user B with 50 total reviews and only 2 of those about Mexican restaurants.

- **Average review length in C**: The average length for the reviews in category C, and it can again be used either as an absolute number or a boolean.

- **Average rating in C** The average of star ratings for all the reviews in category C written by the user. Along with the overall average rating, it is a good indication of how the user looks at this category. If the average rating of the user for ”museums” is much lower than her average rating overall, then we can say that maybe she doesn't like visiting museums.

- **Variance in ratings in C**: The variance of the star ratings for reviews in category C, helps make a better sense of the average.

- **Votes**: This is the number of votes the user receives on the reviews she has written in category C. Yelp gives users three options to vote: ”useful”, ”funny”, and ”cool”, which are very clear indicators of how others benefit from the user's reviews, which is a very important quality for a candidate local expert. We can also explore the relative numbers of these votes, since a user A with 100 votes, with only 10 useful, and 90 funny, might not have equally good knowledge as a user B with just 30 votes, but 25 of them being useful votes.

- **Number of unique businesses**: This is the number of unique businesses in category C that have been reviewed by the user. It denotes the spread of the user's knowledge. User A

who visited the same Indian restaurant a 100 times and wrote 60 different reviews about it, is much less of an expert in Indian food than User B, who visited 30 different Indian restaurants in the area, and reviewed each of them only once.

### 3.5.3 User bias features

This set of features tries to understand how the user's ratings compare to other users' ratings and thus helps mitigate the bias where some users may always give good ratings compared to some other users.

- **Difference in rating**: This is a very simple score that is the absolute difference between : a) the average of all the ratings that the user gave to restaurants in C, and b) the average of the business ratings (average of the all the ratings that the business got from any user) of all those businesses in category C that were reviewed by the user.

- **Difference in rating distributions**: This is a more sophisticated representation of how the users opinions in category C differ from the prevalent opinion of all the users in category C. To get this measure, we take two histograms and then compare them to each other using Kullback-Leibler divergence, one of the standard methods [14] to compare two probability distributions.

    1. The distribution of star ratings given by the user for businesses in category C, with a granularity of half-stars. It is a distribution of the number of businesses that were given 0.5, 1, 1.5, ..., 5 star ratings by the user.
    2. The distribution of the average (average of ratings by all users) star ratings for the same restaurants.

    For discrete probability distributions P and Q, the Kullback Leibler divergence of Q from P is defined to be

$$D_{KL}(P \parallel Q) = \sum_{i} P(i) ln \frac{P(i)}{Q(i)}$$

We use all or a subset of these features to train a classifier and then use this classifier to predict whether the user has topical authority or not.

# Chapter 4

# Experiments and Results

To test and analyze the algorithms processed, we performed some experiments on the dataset. As we mentioned earlier, there is a huge number of categories that Yelp annotates the businesses with. The tables table 4.1 and 4.2 give the top 10 categories, and the top 10 categories related to food and shopping, respectively. Not very surprisingly, restaurants, shopping, and food are the top three categories.

| **Category** | **No. of businesses** |
|---|---|
| Restaurants | 14303 |
| Shopping | 6428 |
| Food | 5209 |
| Beauty & Spas | 3421 |
| Nightlife | 2870 |
| Bars | 2378 |
| Health & Medical | 2351 |
| Automotive | 2241 |
| Home Services | 1957 |
| Fashion | 1897 |

Table 4.1: Top 10 business categories in Yelp data

Since there are so many categories, and to find local experts, we need to do a category-based analysis, and we want to extract category-specific features, it is not feasible to train a single classifier that decides whether the user is an expert in each of those categories. At the same time, features for a different category would not really be useful for finding experts in one category. For example, if

| Category | No. of businesses |
|---|---|
| Mexican | 1749 |
| American (Traditional) | 1508 |
| Fast Food | 1488 |
| Pizza | 1449 |
| Nightlife | 1338 |
| Sandwiches | 1336 |
| Bars | 1258 |
| Coffee & Tea | 1198 |
| American (New) | 1058 |
| Italian | 1008 |

(a) Top food categories

| Category | No. of businesses |
|---|---|
| Fashion | 1897 |
| Home & Garden | 833 |
| Women's Clothing | 632 |
| Beauty & Spas | 450 |
| Department Stores | 441 |
| Cosmetics & Beauty Supply | 430 |
| Food | 403 |
| Sporting Goods | 397 |
| Drugstores | 390 |
| Accessories | 375 |

(b) Top shopping categories

Table 4.2: Top food and shopping categories in Yelp data

we are trying to find experts in local coffee shops, the number of reviews that the user has written about local hospitals is not a good feature. Considering these points, we decided to train an expert model for one category at a time. The following sections present the results for a couple of those categories. But first, we talk about how we generated the positive and negative samples.

## 4.1 Elite Members: Class Labels

Since there is no direct indication in Yelp user data about them being experts, I use the "elite" field to get the ground truth. Yelp has an elite user program where users can apply to become an elite member, and Yelp approves people it thinks are active and influential enough in their locality. This is a perfect field to use for the ground truth data, and generate class labels for the classifier. The elite field is a list of years when the user was an elite member, and we pick all the users that have ever been elite members as positive examples for experts. Out of the 252898 total users, 20045 have been an elite member for atleast one year.

## 4.2 Category Expertise Experiments

We described a large number of features that can be used to train the classifier that tells whether the user is an expert in that category or not. For this work,we used some different subsets of those feature to train the classifier and then analyze how they perform.

1. **Feature set 1**: This feature set includes the user-specific features and the star-rating based ones.

    - Total reviews : Total number of reviews written by user
    - Category reviews : Number of reviews written by user for businesses in the category
    - Average rating by user for businesses in the category
    - Standard dev of ratings by user for businesses in the category
    - Funny : Number of funny votes user got for all her reviews in this category
    - Useful : Number of useful votes user got for all her reviews in this category
    - Cool : Number of cool votes user got for all her reviews in this category
    - Number of unique business in category reviewed by user
    - Months : Number of months for which user has been yelping

2. **Feature set 2**: This includes feature set 1 plus:

    - Number of friends the user has on Yelp

    which gives a sense of how connected and influential the user is on Yelp.

3. **Feature set 3**: This includes feature set 1 plus:

    - Long review: 1 if average length of review in the category is greater than a threshold (set to 20 for these experiments), and 0 otherwise

which is a simple way to include the actual review text written by the user into the classifier.

4. **Feature set 4**: This is a superset of feature sets 1, 2, and 3.

We used the scikit-learn [8] library of python to train three different types of classifiers: Naive Bayes, decision trees, and random forests. Below are the prediction accuracies for all possible combinations of the 4 feature sets and the 3 classifiers for two categories: Chinese food restaurants, and Fashion shopping outlets. The accuracies are averages of 4 runs using randomly picked 33.33 percent of the data as test data and the other 66.67 percent as training data.

### 4.2.1 Chinese restaurants

These are the businesses that have both ”Chinese” and either ”Food” or ”Restaurants” in their categories list. There are 1002 such Chinese food restaurants in the Yelp academic dataset, that have gathered more than 38000 reviews over the years. There are 186 users who have written more than 10 reviews for these restaurants, and about 50 users who have written more than 20 reviews. Table 4.3 gives the prediction accuracies for this category.

| **Classifier** | **Feature set 1** | **Feature set 2** | **Feature set 3** | **Feature set 4** |
|---|---|---|---|---|
| **Naive Bayes** | 0.889 | 0.892 | 0.891 | 0.893 |
| **Decision Trees** | 0.902 | 0.903 | 0.902 | 0.904 |
| **Random Forests** | 0.927 | 0.928 | 0.928 | 0.926 |

Table 4.3: Prediction scores for chinese restaurants

### 4.2.2 Fashion shopping outlets

These are the businesses that have both ”Shopping” and ”Fashion” in their categories list. There are 1897 such businesses in the Yelp academic dataset, that have gathered more than 17880 reviews over the years. Table 4.4 gives the prediction accuracies for this category.

| Classifier | Feature set 1 | Feature set 2 | Feature set 3 | Feature set 4 |
|---|---|---|---|---|
| **Naive Bayes** | 0.840 | 0.848 | 0.843 | 0.848 |
| **Decision Trees** | 0.903 | 0.903 | 0.903 | 0.903 |
| **Random Forests** | 0.926 | 0.928 | 0.927 | 0.926 |

Table 4.4: Prediction scores for fashion outlets

## 4.3 Local Experts

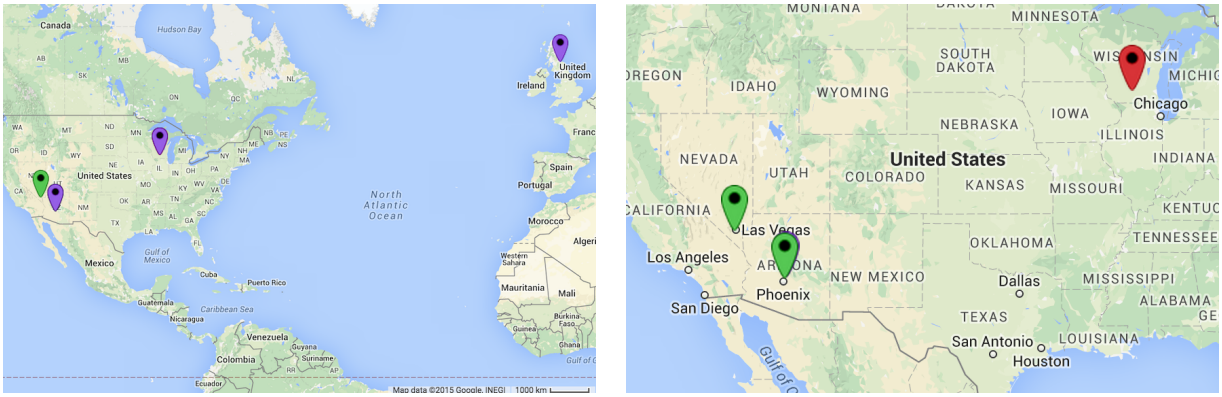
Till now, the results presented just give an idea of how well the topical authority classifiers work, but we have not yet actually taken the location aspect into account. Next we present examples of some actual local queries and the local experts found by the system for those queries.

For this analysis, we took one of our previous categories: fashion outlet businesses, and used a random forest classifier to get a set of candidate experts. Then we pick three of these users to show how the algorithm will pick different users as experts for different queries, based on the location of the query. Table 4.5 shows the location model centers for these users found using GMM, and the last column is a legend of the colors used to mark these on the map in Figure 4.1, which shows these user center locations marked on a map. In Fig. 4.1(a), some of the markers are hidden due to overlapping, and thus Fig. 4.2(b) shows the map zoomed in on the US, showing only the first two users, user1 and user2.

| User No. | Location centers | color |
|---|---|---|
| user1 | 43.07587141,-89.46799598 | Red |
| user2 | 36.09907186,-115.18811982<br>33.54679089,-112.00830908 | Green |
| user3 | 43.08844492,-89.40708842<br>55.9536252,-3.1932316<br>33.6296279,-111.909773 | Purple |

Table 4.5: Location centers for fashion outlet experts

Now, we take two queries with two separate locations: one, I took my own location in Mountain View (37.4026130,-122.0807040), and the other query location is somewhere in Edinburgh, London (55.6095872,-4.6802356). Table 4.6 shows the distances (in miles) of both the query lo-




(a) Points for all the three users

(b) Zoomed in on the US

Figure 4.1: Location centers for fashion outlet experts

cations from the three users. The distances are the minimum of the distance of the query location from all the user location centers, as we formulated in the user location authority section in chapter 3.

| **Query** | **User 1** | **User 2** | **User 3** |
|---|---|---|---|
| **Query 1: Mountain View** | 1753.391 | *392.184* | 628.49 |
| **Query 2: Edinburgh** | 3667.715 | 4905.291 | *62.51* |

Table 4.6: User Distances from Query Locations

For this demonstration, we assume that the distance threshold is very big, lets say 400 miles. We can clearly see that for query1, user2 is a local expert, and for query2, user3 is the local expert. This shows that even though all three users are experts on fashion shopping outlets somewhere, we can answer the local expert question differently for different users based on the location query.

# Chapter 5

# Conclusions and Future Work

This work solves the problem of finding local experts from the Yelp dataset. There have been other works that find experts, and even works that even find local experts from datasets. But this work is novel in the sense that it finds local experts from Yelp dataset. Since Yelp is a collection of local businesses, and aims at helping users find the best local businesses for each need, local experts give a better understanding and a great way to analyze the local business scene in each category.

Looking at the experimental results, we can clearly see that all four feature sets perform almost equally well for both the datasets. At the same time, we can see that the random forests algorithm works the best and gives the best prediction accuracy.

## 5.1 Future Work

There are a lot of possibilities for future work in the area of finding local experts from Yelp dataset. We will talk about some of the ideas we have about some future work, and they are in two categories: one is to enhance the feature set to include more comprehensive features, and, the other is to find new ways to use the local experts found and hence, understand the utility of doing so.

### 5.1.1 Review text analysis

Till now, we didn't actually analyze the review text, other than the length, and it is a big source of information. It can be very beneficial to use the review text to generate many more features and

use them for the classifiers. One very simple thing to do would be to do a topic analysis and then use that to get an interpretation of the text. A good feature is to take the distribution of the topics in all the review text written by a user, and compare it to the distribution of all the reviews written by all the users in the same category.

### 5.1.2 Review summarization

Once we have the set of local experts for a category in a location, we can use it to generate a summary for the businesses in that category. The expectation is that since these users have been identified to be knowledgeable and influential, using just their reviews to generate a text summary would give a higher quality summary about a business. This would save users the need to read through all the reviews of a business, and help them make their decisions based on the collective knowledge of the local experts.

Having done that, the same summarization can also be used to analyze the quality of our local-expert finding algorithm. Once we have the summaries, we can do a manual analysis and say that local experts found are good if the summary is found to be good for a majority of the businesses. Local experts can also be used to generate recommendations for people looking for new places to explore.

# References


[1] AirBnb. Airbnb. http://www.airbnb.com, 2008.

[2] Judd Antin, Marco de Sa, and Elizabeth F. Churchill. Local experts and online review sites. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion*, CSCW '12, pages 55–58, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1051-2. doi: 10.1145/2141512.2141541. URL http://doi.acm.org/10.1145/2141512.2141541.

[3] Krisztian Balog, Leif Azzopardi, and Maarten De Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50. ACM, 2006.

[4] Jennifer King Bryan Hood, Victor Hwang. Inferring future business attention. *Yelp Academic Challenge Dataset*, 2012.

[5] Christopher S Campbell, Paul P Maglio, Alex Cozzi, and Byron Dom. Expertise identification using email communications. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 528–531. ACM, 2003.

[6] Zhiyuan Cheng, James Caverlee, Himanshu Barthwal, and Vandana Bachani. Who is the barbecue king of texas?: A geo-spatial approach to finding local experts on twitter. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 335–344, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2257-7. doi: 10.1145/2600428.2609580. URL http://doi.acm.org/10.1145/2600428.2609580.

[7] Eunkwang Joo James Huang, Stephanie Rogers. Improving restaurants by extracting subtopics from yelp reviews. *Yelp Academic Challenge Dataset*, 2012.

[8] Scikit Learn. http://scikit-learn.org/stable/index.html.

[9] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '08, pages 34:1–34:10, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-323-5. doi: 10.1145/1463434.1463477. URL http://doi.acm.org/10.1145/1463434.1463477.

[10] Wen Li, Carsten Eickhoff, and Arjen P de Vries. Geo-spatial domain expertise in microblogs. In *Advances in Information Retrieval*, pages 487–492. Springer, 2014.

[11] Jack Linshi. Personalizing yelp star ratings: a semantic topic modeling approach. *Yelp Academic Challenge Dataset*, 2014.

[12] Xiaoyong Liu, W Bruce Croft, and Matthew Koll. Finding experts in community-based question-answering services. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 315–316. ACM, 2005.

[13] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 165–172, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2409-0. doi: 10.1145/2507157.2507163. URL http://doi.acm.org/10.1145/2507157.2507163.

[14] Ofir Pele and Michael Werman. The quadratic-chi histogram distance family. In *Proceedings of the 11th European Conference on Computer Vision: Part II*, ECCV'10, pages 749–762, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15551-0, 978-3-642-15551-2. URL http://dl.acm.org/citation.cfm?id=1888028.1888085.

[15] Vayable. Vayable. http://www.vayable.com, 2013.

[16] Felix W. On the efficiency of social recommender networks. *Yelp Academic Challenge Dataset*, 2014.

[17] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. Finding similar users using category-based location history. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 442–445, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0428-3. doi: 10.1145/1869790.1869857. URL http://doi.acm.org/10.1145/1869790.1869857.

[18] Jun Zhang, Mark S. Ackerman, and Lada Adamic. Expertise networks in online communities: Structure and algorithms. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 221–230, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242603. URL http://doi.acm.org/10.1145/1242572.1242603.

[19] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 791–800, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526816. URL http://doi.acm.org/10.1145/1526709.1526816.