Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media

Glaucus: Exploiting the Wisdom of Crowds for Location-Based Queries in Mobile Environments

Minsoo Choy and **Jae-Gil Lee**[*] and **Gahgene Gweon** and **Daehoon Kim**
Department of Knowledge Service Engineering
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea
{ minsoo.choy, jaegil, ggweon, daehoonkim }@kaist.ac.kr

Abstract

In this paper, we build a social search engine named *Glaucus* for location-based queries. They compose a significant portion of mobile searches, thus becoming more popular with the prevalence of mobile devices. However, most of existing social search engines are not designed for location-based queries and thus often produce poor-quality results for such queries. Glaucus is inherently designed to support location-based queries. It collects the check-in information, which pinpoints the places where each user visited, from location-based social networking services such as Foursquare. Then, it calculates the expertise of each user for a query by using our new probabilistic model called the *location aspect model*. We conducted two types of evaluation to prove the effectiveness of our engine. The results showed that Glaucus selected the users supported by stronger evidence for the required expertise than existing social search engines. In addition, the answers from the experts selected by Glaucus were highly rated by our human judges in terms of answer satisfaction.

1 Introduction

Social search is a new paradigm of knowledge acquisition that relies on the people of a questioner's social network (Evans and Chi 2008). In social search, a query is processed by finding a right *person* for that query rather than a right *document*. Nowadays social search is getting more attention. Microsoft and Facebook launched a new service, with the slogan "search goes social," that lets Bing users to tap into the wisdom of friends and experts on Facebook. The side bar of Bing now shows a list of friends and their posts about the topic related to a query. In addition, Facebook recently launched a new service called "Graph Search" that enables Facebook users to find the information through their friends and connections.

Social search is more suitable for subjective queries and personal recommendations than existing search engines and Q&A services such as Google and ChaCha (Arrington 2008; Bulut, Yilmaz, and Demirbas 2011; Horowitz and Kamvar 2010). On the other hand, Google and ChaCha are better at answering factual queries that expect definitive answers. Bulut et al. (Bulut, Yilmaz, and Demirbas 2011) reported that their social search engine answered 75% of both the factual and non-factual queries whereas Google answered 78% of the factual queries and only 29% of the non-factual queries. Such advantage of social search sounds reasonable since people tend to trust the opinions from their family or friends more than those from unfamiliar people (Tan et al. 2013).

Meanwhile, as mobile devices such as smart phones become more prevalent, they are being used ubiquitously in our daily life. Many people search for desired information at various locations and times using mobile devices. Currently, mobile search is estimated to comprise 10%∼30% of all searches depending on the target category (Cleave 2012). Even though desktop search is preferred over mobile search at this time, the gap will decrease as mobile devices become more user-friendly, e.g., Apple Siri. In addition, an executive of Google said that "mobile will be the primary way people will access Google and its many services" (Steiber 2012). In fact, Google reported that their mobile searches increased by 200% in 2012.

Popular queries in mobile search include location-based queries, which are defined as "search for a business or place of interest that is tied to a specific geographical location" (Amin et al. 2009). The frequency of such queries was measured by major search engines companies. About 9∼10% of the queries from Yahoo! mobile search (Yi, Maghoul, and Pedersen 2008), over 15% of 1 million Google queries from PDA devices (Kamvar and Baluja 2006), and about 10% of 10 million Bing mobile queries (Song et al. 2013) were related to local services, which require location information. These location-based queries mostly ask for subjective opinions. A study showed that, in a set of location-based queries, 63% of them were non-factual, and the remaining 37% of them were factual (Bulut, Yilmaz, and Demirbas 2011).

In this paper, combining these two recent trends (social search and mobile search), we develop a social search engine called *Glaucus*[1] that is specialized for location-based queries. Here, the *location-based queries* are selected as the target of our research since they compose a significant portion of mobile searches. The popularity of location-based

[*]Jae-Gil Lee is the corresponding author.

[1]Glaucus, a little owl in Greek mythology, is seen as a symbol of wisdom because the owl is capable of seeing even in the dark.

queries is expected to increase as the prevalence of mobile devices increases. In addition, *social search* is chosen as the methodology of our research since it is good at handling the queries asking for subjective opinions, which are common in location-based queries.

Compared with a *general-purpose* social search engine Aardvark (Horowitz and Kamvar 2010), the difference lies in the way of handling location-based queries. Aardvark selects the experts using the location specified in their user profiles. Such locations are typically specified at *city level*, e.g., Santa Clara, but location-based queries contain more specific locations, e.g., "What is the best Korean restaurant *in Lawrence Plaza in Santa Clara*?" In this example, *not* every person living in Santa Clara is familiar with Lawrence Plaza. Thus, general-purpose social search engines have limitations in processing location-based queries precisely.

Overall, Glaucus has two distinct advantages compared with existing social search engines.

1. Glaucus is capable of providing higher-quality answers for *location-based queries* than general-purpose social search engines. To this end, we collect the data of user behaviors from location-based social (or geosocial) networking services such as Foursquare, Facebook Places, and Yelp. Since this kind of social networking services record the locations where each user has visited (i.e., checked-in), the expertise and familiarity on a specific location can be evaluated in a fine granularity. Most important, we develop a formal probabilistic model, which we call the *location aspect model*, to judge the suitability of each user to a given location-based query. An ingredient of a query and user behaviors is called a *topic*, and the topics are categorized into four types—business name (e.g., Starbucks), business category (e.g., Coffee Shop), location name (e.g., Santa Clara), and time (e.g., 7 p.m.)—to better represent the information relevant to location-based queries.
2. Glaucus is designed to support *alternative recommendation* when it does not have enough information to find clear answers for a query. Since most users do *not* check-in at every place which they visited, it is very important to attempt to answer a query to the extent possible with available information. For example, athough no user in the social network has visited Lawrence Plaza, it will be better to route the query to the persons who have frequently visited "SGD Tofu House," which is located across the street from Lawrence Plaza, rather than to nobody. It is highly likely that such persons visited Lawrence Plaza as well but did not check-in there. To this end, all the topics in each type are organized in a hierarchical tree, and the similarity between two different topics is calculated as the distance on the tree. Consequently, even between different topics, the similarity can be high if they are closely related to each other. In this way, we try to maximize the chances of satisfying questioners through alternative recommendation.

To the best of our knowledge, this is the first work on developing a formal model for location-based social search that considers check-in information as well as alternative recommendation. Nevertheless, since this work is the first step toward our final goal, our model is yet to cover all the aspects of location-based social search. The expertise of a user for a query is mainly considered in this paper, and other aspects such as the likelihood of getting an answer within a short period will be studied in our subsequent papers.

2 Related Work

2.1 General-Purpose Social Search

Horowitz and Kamvar (Horowitz and Kamvar 2010) presented a realtime social search engine named *Aardvark*. They designed a model based on the aspect model (Hofmann 1999), which puts a variable named *topic* between a *user* and a *query* and then finds experts who may know a lot about the topics associated with the query. For location-sensitive queries, Aardvark selects the candidate experts whose location information (mostly from their social network profile) matches the target location of a query. Glaucus also uses the aspect model, but its detailed approach is completely different from Aardvark.

Richardson and White (Richardson and White 2011) developed a synchronous social Q&A system called *IM-an-Expert*. For each user, the system maintains a topic list that represents the user's interests from explicit sources (e.g., his/her user profile page) and implicit sources (e.g., his/her mailing list and Q&A history). Then, the system identifies the experts using a TF/IDF-like ranking algorithm. It is not reported how IM-an-Expert handles location-based queries.

These general-purpose social search engines are shown to successfully support some kinds of queries. However, they have limitations in supporting location-based queries, as discussed in Introduction. A comparison between Glaucus and Aardvark will be given in Section 5.2.

2.2 Location-Based Social Search

Bulut et al. (Bulut, Yilmaz, and Demirbas 2011) developed a crowdsourcing system for location-based queries. To find experts, the system utilizes the bio information of Twitter users and selects the users who live in the city that a query contains. In addition, it uses the information of the mayorship[2] of Foursquare when they have a Foursquare account linked to their Twitter account. Compared with this system, we use the information of finer granularity, i.e., *individual* visits rather than mayorship, to achieve higher accuracy.

Shankar et al. (Shankar et al. 2012) presented a location-based service *SocialTelescope* that makes use of mobile social network interactions. Please note that this service tries to find places just like Google Place Search, not persons. The places are ordered by the number of user visits weighted by their expertise for a search keyword. The expertise of a user is defined as the fraction of times the user has visited any place that matches the keyword. Although SocialTelescope considers individual visits, it does *not* support alternative recommendation owing to simple keyword matching as opposed to Glaucus, as will be shown in Section 5.2.

[2]The mayor of a venue is the user who has visited the venue the most times over the last 60 days.

Church et al. (Church et al. 2010) developed a social search system *SocialSearchBrowser* featured with a map-based interface. Users can see the location-based queries of peers or pose queries of their own on the map. The queries are filtered by the time when a query was submitted, the level of friendship between the user and a questioner, and the similarity to the queries that the user has entered. The goal of the paper is different from ours since it concentrates on interface design and query filtering, not expert finding.

2.3 Question Routing

Another related field is question routing (or expert finding) in community-based question answering (CQA) services. The methods in this field attack the problem of pushing the right questions to the right persons to obtain high-quality answers quickly. The expertise of a user for a query is calculated by not only *textual* similarity between the new query and the previous queries the user answered but also the quality of those answers (Li and King 2010; Zhou et al. 2009). Since these methods typically focus on text data, they are not suitable for location-based queries. A recent technique, developed for Google Confucius (Si et al. 2010), builds a weighted graph between a questioner and answerers and then runs the algorithm HITS on the graph in order to find expert users. The characteristics of the graph, however, may not apply to social search engines because a questioner can connect any user in CQA services but only selected users in social search engines. Thus, the graph-based technique is not directly applicable to Glaucus.

3 System Architecture

3.1 Main Components

In this paper, we develop a social search engine specialized for location-based queries. Figure 1 shows the major steps performed in our social search engine. First, a questioner submits a location-based query to Glaucus using the Android app as in Figures 2(a) or 2(b). The quick questions are templates, and the meaning of N, C, L, T, and P will be explained later. Second, the query submitted is analyzed by the query engine of Glaucus, and k users in the social network are selected as the experts on the query. Third, the query is routed to these experts, and they are notified by the Android app as in Figure 2(c). Fourth, if one of the experts responds to the query, his/her answer is sent back to the questioner through the search engine. Last, the questioner optionally evaluates the helpfulness of the answer, and this feedback is stored in the user database.

The scope of this paper is the second step of Figure 1. That is, the evaluation of the expertise of a user is discussed in the paper. Our social search engine will be extended to consider other factors such as availability.

3.2 User Database

The user database stores the information required for evaluating the expertise of each user for a given location-based query, and it is populated by crawling the data from location-based social (geosocial) networking services. We collect two types of information: check-in records and review records.

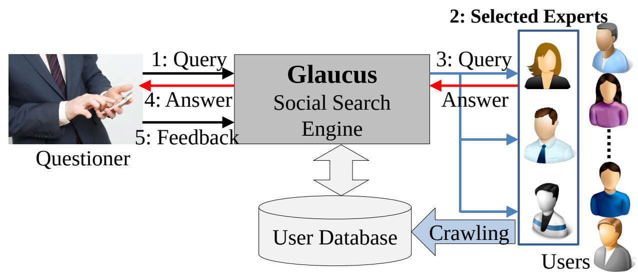
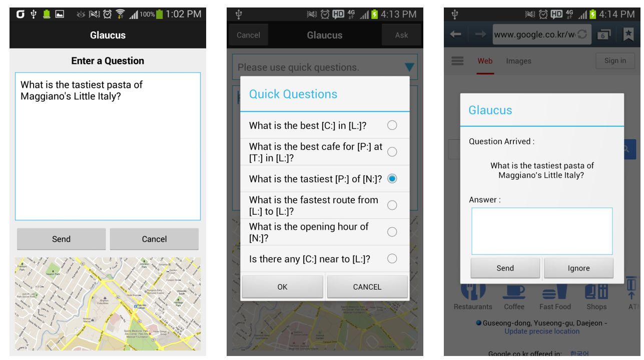


Figure 1: The system architecture of Glaucus.



(a) Free Question. (b) Quick Question. (c) Notification.

Figure 2: Screen shots of the prototype of Glaucus.

Check-in on a social networking service is the process whereby a person announces his/her arrival at a certain place. Many social networking services such as Foursquare, Facebook Place, and Google+ allow users to check-in to a physical place and share their locations with their friends. This check-in information should be very useful for processing location-based queries because it directly indicates the expertise of a user in a specific place. The more frequently the user has checked-in to a place, the more expert the user is on the place. A piece of the check-in information is called a *check-in record* and defined in Definition 1.

Definition 1. A *check-in record* is $\langle user\_id, place\_id, timestamp\rangle$, which means that $user\_id$ went to $place\_id$ at $timestamp$. □

In social networking services, users often write a review (or a tip) on a specific place, product, or service. This review information is complementary to the check-in information since it reveals a user's taste or preference at the place where the user checked-in. A piece of the review information is called a *review record* and defined in Definition 2.

Definition 2. A *review record* is $\langle user\_id, place\_id, content, timestamp\rangle$, which means that $user\_id$ wrote $content$ on something about $place\_id$ at $timestamp$. □

3.3 Queries and Topics

A *location-based query* is informally defined in Definition 3, which basically expresses location-based information needs. This definition is quite broad in considering that it actually implies a class of questions that ask for any type of information involving locations.

Definition 3. A *location-based query* is defined as "search for a business or place of interest that is tied to a specific geographical location" (Amin et al. 2009). □

Example 1. Location-based queries often occur in daily life. Just a few examples are (i) "What is the best Korean restaurant in Lawrence Plaza in Santa Clara?", (ii) "What is the best cafe for brunch on the weekend at Santana Row?", and (iii) "What is the tastiest pasta of Maggiano's Little Italy?" □

A location-based query usually comprises multiple components, and each component is called a *topic*, which is defined in Definition 4. Consequently, a review/check-in record is decomposed into multiple topics.

Definition 4. A *topic* is the smallest piece of information contained in location-based queries and check-in/review records. □

The topics in a location-based query tend to be a combination of multiple types of information. For example, a query can be a combination of a business category and a location name; another query a combination of a business name and a location name. A similar idea that assigns multiple attributes to an object is being used in *faceted search*. Thus, the topics had better be classified into multiple categories, as defined in Definition 5.

Definition 5. A *topic category* is a group of topics that represent the same type of information needs. □

As discussed in Appendix A, we decided to use the four topic categories that occur most frequently plus *potpourri* (*P*). Similar categorization can also be found in other studies (Amin et al. 2009; Bulut, Yilmaz, and Demirbas 2011). The categorization of topics depends on the location-based queries in consideration, and our methodology is orthogonal to this categorization.

- *business name* (N): e.g., Starbucks, Olive Garden
- *business category* (C): e.g., Coffee Shop, Thai Restaurant
- *location name* (L): e.g., Santa Clara, Palo Alto
- *time* (T): e.g., evening, late night
- *potpourri* (P): those do not belong to the above

Example 2. The topics of the queries in Example 1 are categorized as below. Here, a capital letter in square brackets denotes the corresponding topic category.

(i) "What is the [**P**:best] [**C**:Korean restaurant] in [**L**:Lawrence Plaza] in [**L**:Santa Clara]?"
(ii) "What is the [**P**:best] [**C**:cafe] for [**P**:brunch] on the [**T**:weekend] at [**L**:Santana Row]?"
(iii) "What is the [**P:**tastiest] [**P**:pasta] of [**N**:Maggiano's Little Italy]?" □

The topics are extracted from natural-language queries in Figure 2(a) with help from named-entity recognition (NER). On the other hand, topic extraction is straightforward for quick questions in Figure 2(b).

3.4 Expert Finding

Expert finding in Glaucus, which we want to solve in this paper, is stated in Definition 6.

Definition 6. *Expert finding* in Glaucus is, given a set of users $\mathbb{U}$, a set of check-in records $\mathbb{C}$, a set of review records $\mathbb{R}$, a questioner $u_q \in \mathbb{U}$ who asked a location-based query q, and a parameter k, to derive the expertise score $score(u_i, u_q, q)$ for each user $u_i \in \mathbb{U}$ and then to return the top-k users according to the scores. □

Figure 3 shows the overall procedure of our expert finding. The key idea is to combine the scores calculated *separately for each topic category*. This approach makes sense because the topics of different topic categories are *not* simply comparable. Thus, we contend that comparison within a topic category should lead to more precise measurements of the relevancy between users and a query.

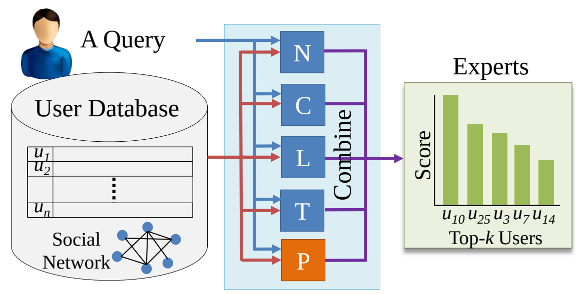


Figure 3: The overall procedure of expert finding.

Our expertise score basically reflects how much the topics from a user match those from a query. That is, the topics can be considered as *the bridges between users and a query*. The topics are first extracted from the check-in records and the review records, representing user expertise. In addition, the topics are extracted from a query, representing the questioner's intention. Then, for each topic category, the topics from the query are compared against those from each user to derive the expertise score of the user for that query.

The relationships among users are also utilized to give more trust to the answers from intimate friends than those from unfamiliar users. This special treatment of intimate friends is due to the philosophy of social search.

4 Location Aspect Model

4.1 Overview of the Model

We now elaborate on how to calculate the expertise score of each user for a given query in the location aspect model. As its name indicates, it is an extension of the aspect model (Hofmann 1999) also used in the Aardvark search engine. Please note that the aspect model is a quite general model, which has been widely used in information retrieval. Aardvark and Glaucus are completely different regarding how to use the aspect model. Our new model is specialized for location-based queries.

In our model, the expertise score $score(u_i, u_q, q)$ is defined in Eq. (1). Here, the three components are defined as follows.

$$score(u_i, u_q, q) = p(u_i|u_q) \cdot combine(p(u_i|q), boost\_factor(u_i, q)) \tag{1}$$

- $p(u_i|u_q)$: the probability that the questioner u_q will be satisfied by the user u_i's answer (See Section 4.2)
- $p(u_i|q)$: the probability that the user u_i will successfully answer the question q (See Section 4.3)
- $boost\_factor(u_i, q)$: the degree of boosting based on the match between the preference of the user u_i and the potpourri topics of the query q (See Section 4.4)

$p(u_i|u_q)$ is query-independent, so it can be calculated in advance using the social network of u_q. On the other hand, $p(u_i|q)$ and $boost\_factor(u_i, q)$ are query-dependent. $p(u_i|q)$ is calculated using the similarity between the topics in N, C, L, and T respectively. For the topics in P, it is hard to define similarity since the topic category may contain various semantics. Thus, instead of calculating a probability formally, we decided to simply boost the probability $p(u_i|q)$ to a small extent by $boost\_factor(u_i, q)$ calculated using the topics in P.

4.2 Similarity between Users

$p(u_i|u_q)$ is calculated by Eq.(2). Here, $\mathbb{F}$ denotes the set of the questioner's friends in the social network. Just for simplicity of the model, the value of $p(u_i|u_q)$ can be one of the two values. Suppose that a user u_j is a friend of u_q whereas u_k is not. Then, $p(u_j|u_q)$ is w_{friend} times higher than $p(u_k|u_q)$. This component is to reflect the fact that people tend to trust the opinions from their family or friends more than those from unfamiliar people (Tan et al. 2013). w_{friend} is a tuning parameter and depends on the context.

$$\forall u_j \in \mathbb{F}, u_k \in \mathbb{U} - \mathbb{F}, p(u_j|u_q) = p(u_k|u_q) \cdot w_{friend} \\ \text{such that} \sum_{u_i \in \mathbb{U}} p(u_i|u_q) = 1 \tag{2}$$

We divide the entire set of users into friends and non-friends of a questioner. Only the friends are considered to be close to the questioner. One might think that the friends of friends had better be considered since they can reach with only two hops in the social network. However, many social networking services continuously expose the friends of friends and encourage us to make direct friendships with them. If they still remain as non-friends, it is likely that they are *not* really close to the questioner.

4.3 Similarity between Users and Queries

$p(u_i|q)$ is defined by Eq. (3), which means the familiarity of a user u_i to the topics contained in a query q. The three components $p(u_i|t)$, $p(t|q)$, and w_{cat} are described as below. The final value of $p(u_i|q)$ is a weighted sum of the values derived for each topic category. The weight w_{cat} is proportional to the frequency of the corresponding category in location-based queries.

$$p(u_i|q) = \sum_{cat \in \{\text{N,C,L,T}\}} w_{cat} \cdot \sum_{t \in T_{cat}} p(u_i|t) \cdot p(t|q) \tag{3}$$

- $p(u_i|t)$: the probability of a user u_i being an expert for a topic t, as will be defined in Eq. (4)
- $p(t|q)$: the probability of a topic t matching a query q, as will be defined in Eq. (6)
- w_{cat}: a weight (importance) of each topic category in location-based queries (See Appendix A)

Similarity between Users and Topics $p(u_i|t)$, which is the first component of Eq. (3), is defined by Eq. (4). Here, $p(u_i|t)$ is represented using $p(t|u_i)$ by Bayes' theorem. Please note that $p(t|u_i)$ can be calculated directly from the check-in records.

$$p(u_i|t) = \frac{p(t|u_i) \cdot p(u_i)}{p(t)} \tag{4}$$

The check-in records are loaded into the main memory in the form of a *topic-frequency table*. Table 1 shows an example of the topic-frequency table. A row of this table represents a user's behaviors, and a column a topic category. Each entry of the table has a set of key-value pairs, and thus the table does not comply to the first normal form. Here, the key is a topic, and the value in parenthesis is the frequency of the topic. In every column, the frequency of a topic will increase by 1 whenever a check-in record is added.

Table 1: An example of the topic-frequency table.

	location name (L)	business category (C)	time (T)	business name (N)
u_1	Sinsa-dong (20)	Italian Restaurant (20)	Weekday Lunch (20)	Nilly Pasta & Pizza (10) Black Smith (10)
u_2	Sinsa-dong (20) Apgujeong1-dong (10)	Italian Restaurant (30)	Weekday Lunch (20) Weekend Lunch (10)	Black Smith (20) Noah (10)
u_3	Nonhyeon1-dong (20)	Italian Restaurant (20)	Weekday Dinner (20)	Italian Kitchen (15) The Plate (5)
u	Nonhyeon1-dong (15)	Italian Restaurant (15)	Weekend Lunch (5) Weekend Dinner (10)	Italian Kitchen (15)
u_5	Sinsa-dong (15) Apgujeong1-dong (20)	Chinese Restaurant (20) Dessert Shop (15)	Weekend Lunch (35)	Dowon (20) Miltop (15)

Example 3. A check-in record $\langle u_1$, Black Smith, 12:30 p.m. May 20 (Monday)$\rangle$ will increase the frequency of "Black Smith" in the business name category and that of "Weekday Lunch" in the time category. The location name and business category of the place are registered in the location-based social networking service. Thus, the corresponding topics, "Sinsa-dong"[3] and "Italian Restaurant" respectively, will have the frequency increased by 1. □

We now formally explain the derivation of $p(u_i|t)$ using the notation of Table 2. Eq. (4) is translated as Eq. (5) by rewriting $p(t|u_i)$, $p(u_i)$, and $p(t)$. $p(t|u_i)$ is normalized such that $\sum_{t \in T_{cat}} p(t|u_i) = 1$ for each topic category.

$$p(u_i|t) = \frac{\frac{\nu_{topic}(t,u_i)}{\nu_{visit}(u_i)} \cdot \frac{\nu_{visit}(u_i)}{\nu_{visit}}}{\frac{\nu_{topic}(t)}{\nu_{visit}}} = \frac{\nu_{topic}(t, u_i)}{\nu_{topic}(t)} \tag{5}$$

Example 4. Let us show a few examples of calculating $p(u_i|t)$ in Table 1: $p(u_2|$"Sinsa-dong"$) = \frac{20}{55}$, $p(u_2|$"Italian Restaurant"$) = \frac{30}{85}$, $p(u_2|$"Weekend Lunch"$) = \frac{10}{50}$, and $p(u_2|$"Black Smith"$) = \frac{20}{30}$, and so on. □

[3]A *dong* is the smallest level of urban government to have its own office and staff in Korea.

Table 2: The notation for deriving $p(u_i|t)$.

Symbol	Description
$\nu_{topic}(t, u_i)$	the frequency of the topic t in the user u_i
$\nu_{topic}(t)$	the frequency of the topic t in all users
$\nu_{visit}(u_i)$	the number of visits of the user u_i
ν_{visit}	the total number of visits by all users

Similarity between Topics and a Query $p(t|q)$, which is the second component of Eq. (3), is defined by Eq. (6). It basically represents how much a topic in a query q is similar to an existing topic t for each topic category. t^q_{cat} denotes the topic contained in q with regard to a topic category cat. For ease of explanation, Eq. (6) is formulated under the assumption that a single topic exists in q for each topic category. The equation can be easily extended to handle multiple topics. $p(t|q)$ is normalized such that $\sum_{t \in T_{cat}} p(t|q) = 1$.

$$p(t|q) = \frac{sim_{cat}(t, t^q_{cat})}{\sum\limits_{t_i \in T_{cat}} sim_{cat}(t_i, t^q_{cat})} \tag{6}$$

Our derivation of $p(t|q)$ is the core of supporting *alternative recommendation*. Although no existing topic exactly matches a topic in a query, we try our best to find the topics that are very "similar" to the topic of the query. Please recall our example considering "SGD Tofu House" instead of Lawrence Plaza. Finding such alternatives becomes possible by carefully designing a similarity function between topics.

The similarity function should reflect the characteristics of a topic category as illustrated in Figure 4.

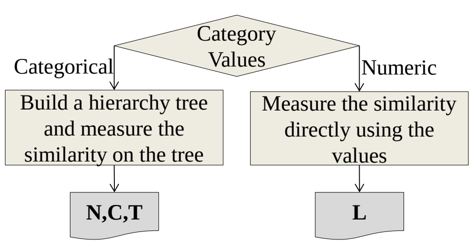


Figure 4: The approach to calculating the similarity between topics.

1. **business name (N), business category (C), time (T)**: The similarity function is based on the distance on the hierarchy tree. The *least common ancestor* or *least common superconcept* (Wu and Palmer 1994) of the nodes n_i and n_j is the lowest node in the tree that is an ancestor of both n_i and n_j. If n_{lca} is the least common ancestor of n_i and n_j, then $dist_{tree}(n_i, n_j)$ is the number of steps from n_i to n_{lca} plus the number of steps from n_j to n_{lca}. Then, the similarity function $sim_{cat}(t_i, t_j)$ is defined by Eq. (7). Here, n_{t_i} and n_{t_j} are the nodes corresponding to the topics t_i and t_j respectively. We apply the exponential decay function since exponential decay in similarity as a function of distance is observed frequently in natural sciences (Poulin 2003).

$$sim_{N|C|T}(n_{t_i}, n_{t_j}) = e^{-dist_{tree}(n_{t_i}, n_{t_j})} \tag{7}$$

For business category (C), a hierarchy tree was adopted from the nested lists of business categories of a social networking service[4]. For business name (N), we decided to share the same tree after inserting each place as a child of its business category.

For time (T), a hierarchy tree was built by analyzing the check-in records we collected. Since user behaviors are quite different between during the week and on the weekend, the tree is first split on this condition. Then, a day is split into six ranges: the number of check-in's is very low in the morning (6 a.m.$\sim$11 a.m.), increases at lunch time (11 a.m.$\sim$2 p.m.), drops again after lunch time (2 p.m.$\sim$6 p.m.), increases again at dinner time (6 p.m.$\sim$8 p.m.), keeps high from different types of places at night (8 p.m.$\sim$11 p.m.), and starts to drop after midnight.

2. **location name (L)**: The similarity function is defined by Eq. (8). Its values are normalized to be between 0 and 1. Here, $dist_{loc}(\cdot,\cdot)$ returns the distance in kilometers between two points specified by latitude and longitude, and max_{dist} is the maximum distance between registered places or the maximum distance considered to be geographically relevant.

$$sim_L(l_{t_i}, l_{t_j}) = 1 - \frac{dist_{loc}(l_{t_i}, l_{t_j})}{max_{dist}} \tag{8}$$

4.4 The *Potpourri* Topic Category

$boost\_factor(u_i, q)$ of Eq. (1) is calculated using the potpourri (P) topic category, which measures the degree of how much a user u_i's interests match the intention of a query q. A set of keywords are extracted from the review records written by each user u_i. Also, the keywords marked as P in a query q form another set of keywords. The former is denoted by K_{u_i}, and the latter by K_q. $boost\_factor\ (u_i, q)$, which is defined by Eq. (9), increases as the number of pairs of similar keywords from both sets does. The logarithm is applied to avoid the dominance of this topic category.

$$boost\_factor(u_i, q) = \log(|K_{sim}| + 1) \tag{9}$$
$$K_{sim} = \{(k_i, k_j) | \forall k_i \in K_{u_i}, k_j \in K_q, k_i \text{ and } k_j \text{ are similar}\}$$

Two topics (keywords) are determined to be similar by *string similarity*. We simply adopted string similarity since it is hard to determine semantic difference between two keywords of diverse semantics. The Dice coefficient (Dice 1945) using n-grams is a widely-used measure for string similarity, which is defined by Eq. (10). Here, $n\text{-}grams(k)$ is a multi-set of letter n-grams in a keyword k. Using *bi-grams* (i.e., $n = 2$) is particulary popular, and this Dice coefficient is usually denoted by $DICE$. For example, $DICE$(*Zantac*, *Contac*) = $(2 \times 3)/(5 + 5)$ = 0.6. We consider that two keywords are similar if $DICE(\cdot,\cdot)$ is greater than or equal to 0.5.

$$\frac{2 \times |n\text{-}grams(k_i) \cap n\text{-}grams(k_j)|}{|n\text{-}grams(k_i)| + |n\text{-}grams(k_j)|} \tag{10}$$

[4]http://aboutfoursquare.com/foursquare-categories/

4.5 Incorporation of All Topic Categories

Once we calculate $p(u_i|q)$ and $boost\_factor(u_i, q)$, we need to incorporate them into a unified score using the $combine(\cdot, \cdot)$ function, which is defined by Eq. (11). The two values of a user u_i are transformed to the proportion of the sum of the values for all users. We finally get a weighted sum of these transformed values. w_{visit} is a tuning parameter and depends on the context.

$$combine(p(u_i|q), boost\_factor(u_i, q)) =$$
$$w_{visit} \frac{p(u_i|q)}{\sum\limits_{u_j \in \mathbb{U}} p(u_j|q)} + (1 - w_{visit}) \frac{boost\_factor(u_i|q)}{\sum\limits_{u_j \in \mathbb{U}} boost\_factor(u_j|q)} \quad (11)$$

5 Evaluation

We conducted two types of evaluation on Glaucus. In Section 5.2, we confirmed the qualification of the selected experts by looking into their check-in/review records. In Section 5.3, we conducted a user study to verify the quality of the answers solicited from the selected experts.

5.1 Experiment Setting

Data Set We collected the check-in records and the review records from Foursquare during the period from April 2012 to December 2012. Only the places (venues) located in the Gangnam District[5] were collected for the experiments. Gangnam is one of the most popular spots in Seoul, Korea and is always very crowded. Since the same place may have duplicate entries in Foursquare, we merged these duplicate places into one identifier by manually investigating all the registered places. Then, we eliminated insignificant places that were visited by less than ten users or less than 100 times. Our crawler collected the check-in's at the remaining places by periodically calling the `herenow` API as well as the reviews (tips) written by the users who visited there. As a result, our data set was obtained as in Table 3.

Table 3: The statistics of our real data set.

Variable	Value
# of users	9,163
# of places (venues)	1,220
# of check-in records	243,114
# of review records	40,248

Compared Systems We compared Glaucus with two existing systems: Aardvark (Horowitz and Kamvar 2010) and SocialTelescope (Shankar et al. 2012). The two systems were re-implemented by us as below.

- *Aardvark*: Since the target location of our experiments was Gangnam in Seoul, Aardvark first selected the users whose current location was specified as Seoul in their profile page, considering only 2,364 out of 9,163 users. Here, we consulted Facebook accounts linked to Foursquare accounts since Foursquare itself does not maintain current locations. Then, Aardvark extracted topics from the posts in social networking services. These posts correspond to the review records in our data set. Similarity between extracted topics is calculated using corpus-based semantic similarity. Since a specific similarity measure was not mentioned, we adopted the *PMI-IR* measure (Mihalcea, Corley, and Strapparava 2006), which has been widely used in information retrieval.
- *SocialTelescope*: Although SocialTelescope is designed for venue recommendation, it does calculate the expertise of users by considering the number of their visits to the places *exactly* matching a search term. When a query consists of a single term, the expertise of a user u_i for a query q is defined by Eq. (12) using the notation of Table 2. When a query consists of multiple terms, we sum up the scores for each term. This design emphasizes the users who visited *only the matching places* because it includes the *ratio* of the number of such visits to the total number of visits, i.e., $\frac{\nu_{topic}(q,u_i)}{\nu_{visit}(u_i)}$.

$$score(u_i, q) = \frac{\nu_{topic}(q, u_i)}{\nu_{visit}(u_i)} \cdot \log \frac{C}{\nu_{topic}(q)} \quad (12)$$

Location-Based Queries For comparison with existing systems, we selected 30 queries from the set of location-based queries in Appendix A. The detailed results, however, are presented for only two queries Q_{c1} and Q_{c2} owing to the lack of space. Q_{c1} includes the C, L, and P topic categories; Q_{c2} the N, T, and P topic categories. "Miltop" is a famous dessert cafe in Seoul. For user study, we used another set of ten location-based queries.

- Q_{c1}: Which Italian restaurant in Sinsa-dong does serve delicious carbonara?
- Q_{c2}: Is Miltop crowded at lunch time during weekdays?

Parameter Configuration There are only two tuning parameters that need to be configured empirically: w_{friend} of Eq. (2) was set to be 1.4 such that the ranking of a friend was typically improved by up to k (the number of experts to select) compared with that of a non-friend; and w_{visit} of Eq. (11) was set to be 0.7 such that the effect of reviews (tips) did not stand out. As discussed in Appendix A, w_{cat}'s were determined by measuring the frequency of each topic category from 1,100 sample queries.

5.2 Qualification of Selected Experts

Preliminary Comparison with Aardvark Table 4 shows the top-5 experts on Q_{c1}. Each row represents a user, and each column a topic category showing some related topics of the user with their frequency in parenthesis. For example, the user 4548829 visited Sinsa-dong 103 times and Nonhyeon1-dong 13 times. The top-5 rankers of Glaucus are all qualified since they have visited Sinsa-dong and/or Italian restaurants very often. The user 4548829 was ranked at the first in both engines. However, the 2nd$\sim$5th rankers of Aardvark do *not* have convincing evidence that they are experts for the query.

Table 5 shows the top-5 experts on Q_{c2}. The top-5 rankers of Glaucus are again all qualified since they have frequently visited Miltop at lunch time during weekdays. The 1st ranker of Aardvark wrote a review about "ice flakes," the main dish of Miltop. However, we are not sure if he/she is really an expert since he/she did not visit there at all.

Overall, the lesson obtained from the comparison between Glaucus and Aardvark is summarized as follows.

[5] http://en.wikipedia.org/wiki/Gangnam_District/

Table 4: Comparison between Glaucus and Aardvark for a query Q_{c1}.

	Glaucus				Aardvark			
	ID	location name (L)	business category (C)	potpourri (P)	ID	location name (L)	business category (C)	potpourri (P)
1	4548829 **(friend)**	**Sinsa-dong (103)** Nonhyeon1-dong (13) ... (2)	Burger Joint (1) Chicken Joint (4) ... (9)	Carbonara Delicious	4548829 **(friend)**	**Sinsa-dong (103)** Nonhyeon1-dong (13) ... (2)	Burger Joint (1) Chicken Joint (4) ... (9)	Carbonara Pasta Delicious ...
2	12661395 (non-friend)	**Sinsa-dong (24)** Apgujeong1-dong (156) ... (5)	**Italian Restaurant (15)**	-	1673674 (non-friend)	Nonhyeon2-dong (3)	-	Carbonara Pasta Italian Restaurant ...
3	484151 (non-friend)	**Sinsa-dong (635)** Apgujeong1-dong (1)	Wings Joint (1)	-	3404766 (non-friend)	**Sinsa-dong (6)** Nonhyeon1-dong (1)	**Italian Restaurant (1)**	Sinsa-dong Asian Restaurant ...
4	1075843 (non-friend)	**Sinsa-dong (138)** Apgujeong1-dong (3) ... (1)	**Italian Restaurant (6)** Burger Joint (1) ... (3)	Delicious	927082 (non-friend)	**Sinsa-dong (3)** Apgujeong1-dong (2)	-	Sinsa-dong Delicious
5	8473811 **(friend)**	**Sinsa-dong (24)** Apgujeong1-dong (1)	**Italian Restaurant (3)** Burger Joint (1)	Delicious	815783 (non-friend)	Sinsa-dong (20)	Bakery (2)	Sinsa-dong Delicious Pasta ...

Table 5: Comparison between Glaucus and Aardvark for a query Q_{c2}.

	Glaucus				Aardvark			
	ID	time (T)	business name (N)	potpourri (P)	ID	time (T)	business name (N)	potpourri (P)
1	8473811 **(friend)**	**Weekday Lunch (2)** Weekday Afternoon (10)	**Miltop (2)** Dessert Shop (2)	People Crowded	3404766 (non-friend)	**Weekday Lunch (1)** Weekday Afternoon (2)	-	Ice Flake Delicious Place ...
2	1446323 (non-friend)	**Weekday Lunch (1)** Weekday Morning (6) ... (2)	**Miltop (4)** Dessert Shop (4)	People	25718337 (non-friend)	**Weekday Lunch (2)** Weekday Morning (2)	-	Delicious Asian Restaurant Place ...
3	9681634 (non-friend)	**Weekday Lunch (7)** Weekday Afternoon (29) ... (14)	**Miltop (15)** Dessert Shop (33)	-	927082 (non-friend)	Weekday Morning (1)	HANS Patisserie (1)	Delicious Place ...
4	23245085 (non-friend)	**Weekday Lunch (15)** Weekday Morning (13) ... (6)	**Miltop (8)** Dessert Shop (8)	-	12685045 (non-friend)	-	-	Lunch Set Thai Restaurant Place ...
5	19201862 (non-friend)	Weekday Morning (3)	**Miltop (7)** Dessert Shop (7)	-	6211750 (non-friend)	-	-	Delicious Fastfood Place ...

- The check-in information is helpful for processing location-based queries since it *pinpoints* the places where each user has visited. On the other hand, the location information of a user profile is not that helpful for the following reasons. First, the current location is typically specified at city level, thus being too broad. All the rankers of Aardvark live in Seoul, but many of them did not visit our points of interest at all. Second, many users are reluctant to expose their current location or do not maintain their user profile up-to-date. Among the users having a Facebook account in our data set, only 54% of the users made their location information public on Facebook. Third, it is possible that the users in other districts often come to our points of interest, possibly because of business trips or long-distance relationships. Surprisingly, almost all the rankers of Glaucus (except 4548829 and 9681634) do *not* live in Seoul, but they did visit our points of interest very often. Glaucus can catch such experts from other districts.
- *Alternative recommendation* is shown to work well in Glaucus by virtue of our sophisticated similarity calculations. The 1st ranker of Glaucus in Table 4 did not visit Italian restaurants but went to *other categories of restaurants* such as burger joints and chicken joints. Moreover, his/her visits to Nonhyeon1-dong, which is very near to Sinsa-dong, contributed to the high score. In addition, the 5th ranker in Table 5 went to Miltop in the morning of weekdays, which is adjacent to lunch time.

Preliminary Comparison with SocialTelescope Table 6 shows the top-5 experts identified by SocialTelescope for Q_{c1} and their statistics. In general, the total number of visits by each user (i.e., $\nu_{visit}(u_i)$) is not very high, and the number of the visits to the matching places (i.e., $\nu_{topic}(q, u_i)$) tends to be close to the total number of visits. In this way, these rankers of SocialTelescope achieve a high value of $\frac{\nu_{topic}(q,u_i)}{\nu_{visit}(u_i)}$, resulting in a high expertise score. They also can be regarded as the experts on Q_{c1} since they have often visited the places in "Sinsa-dong" and those associated with the reviews containing "Delicious."

Table 6: Top-5 experts for a query Q_{c1} by SocialTelescope.

Rank	ID	$\upsilon_{visit}(u_i)$	$\upsilon_{topic}(q, u_i)$			
			Sinsa-dong	It. Restaurant	Carbonara	Delicious
1	5350775	18	15	7	0	10
2	9317351	15	15	2	1	11
3	7063881	16	16	2	1	11
4	846771	15	14	1	2	10
5	5337987	21	21	4	1	9

Overall, the lesson obtained from the comparison between Glaucus and SocialTelescope is summarized as follows.

- SocialTelescope does not support *alternative recommendation* because it does not consider topic similarity. For example, the 1st ranker of Glaucus cannot be found by SocialTelescope.
- SocialTelescope, by its design, has a difficulty in finding the experts who have a wide spectrum of interests. For example, the 2nd ranker of Glaucus visited Sinsa-dong quite many times (24 visits), but these visits occupied only 12.9% of his/her total visits (185 visits). Therefore, the user was ranked very low (4222nd) by SocialTelescope.
- SocialTelescope does not consider social relationship whereas Glaucus does. For example, none of the users in Table 6 is a friend of the questioner whereas two of the users in Table 4 are. The *intimacy* between a questioner and an answerer is important in the social search paradigm (Evans and Chi 2008).

Method Now we will show the superiority of Glaucus over the existing systems *using more queries*. The 30 location-based queries were randomly distributed to three sets (Set 1, 2, and 3) each with ten queries. For each query, we ran the three systems and obtained the top-5 experts of each system. The check-in/review records of a user were extracted and displayed just like Tables 4 and 5. Then, we shuffled users (rows) and systems (column groups) to hide the ranking of a user and the system which ranked the user. Last, we assigned two human judges to each query set and asked them to rate each user's qualification in 3 scales by referring to his/her check-in/review records. The higher the rating is, the better qualification the user has. Cohen's kappa coefficients for Set 1, 2, and 3 are 0.58, 0.57, and 0.45 respectively, which mean moderate agreement.

To measure the effectiveness of a system, we calculated discounted cumulative gain (DCG). The DCG at a rank position p (in our case, $p = 5$) is defined by Eq. (13), where rel_i is the rating of the expert at a position i.

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i)} \tag{13}$$

Results Figure 5 shows the DCG of the three systems in each query set. Error bars indicate the standard error. For the reasons explained in the preliminary comparison studies, the human judges gave the highest rating to the experts selected by Glaucus. SocialTelescope was shown to outperform Aardvark since the former exploits check-in data too.

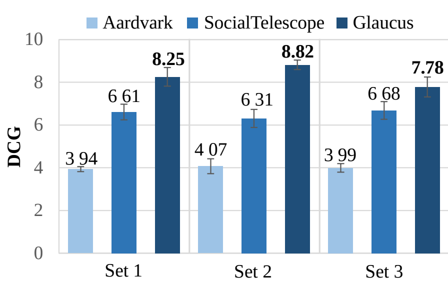


Figure 5: Comparison of the qualification of selected users.

5.3 User Study

Method We conducted a user study to verify that the experts selected by Glaucus are indeed qualified. For each of the ten queries, a set of experts were selected by running Glaucus on the same data set, and a set of non-experts were selected randomly. Then, we sent an e-mail to both experts and non-experts, asking them to participate in an online survey. The survey form solicited for answers to a given location-based query as well as two inquiries: "**[External Information]** *Did you refer to the external information (e.g., Internet) to find the answer to the query?*" and "**[Response Time]** *How long did it take to answer the query?*"

Then, using the survey results, we examined the differences between experts and non-experts in terms of (i) use of external information, (ii) response time, and (iii) answer quality. The equal numbers of responses were taken from the experts and the non-experts. 2$\sim$6 responses were collected per query, and 38 responses were available (19 from the experts and 19 from the non-experts) in total.

To evaluate answer quality, two human judges were requested to review the answers of the 38 responses and give a rating to each answer in 3 scales. The higher the rating is, the better the quality of the answer is. Detailed instructions were provided to the judges for each query. The judges did *not* know whether an answer was from an expert or a non-expert. Cohen's kappa coefficient is 0.82, which means almost perfect agreement.

Results Figure 6(a) shows how many experts and non-experts referred to external information. All the experts answered the queries based on only background knowledge whereas the non-experts did not. Figure 6(b) shows the distribution of the response times from the experts and the non-experts. It is observed that the experts are likely to respond more quickly than the non-experts. Figure 6(c) shows the average rating of the experts' answers and the non-experts' answers. The quality of the experts' answers is higher than that of the non-experts' answers, even though many non-experts referred to external information. The human judges reported that the experts' answers tend to be more detailed and useful than the non-experts' answers. The *p-value* is 0.0013, which means that there is statistically significant difference between the experts and the non-experts.

6 Conclusions

In this paper, we built a social search engine named *Glaucus* for location-based queries. As abundant amounts of check-in and review data are being created on location-based social networking services, our design choice is to take advantage of such data for modeling *locational* user behaviors. Our *location aspect model* formally defines the expertise of each user based on his/her previous activities represented by the check-in and review data. In this model, similarity between topics is measured separately for each topic category in order to better reflect the characteristics of such queries. Another important advantage is that Glaucus supports *alternative recommendation* by virtue of our similarity functions considering the topic hierarchy. Overall, we believe that we made a great start for the implementation of a location-based social search engine.

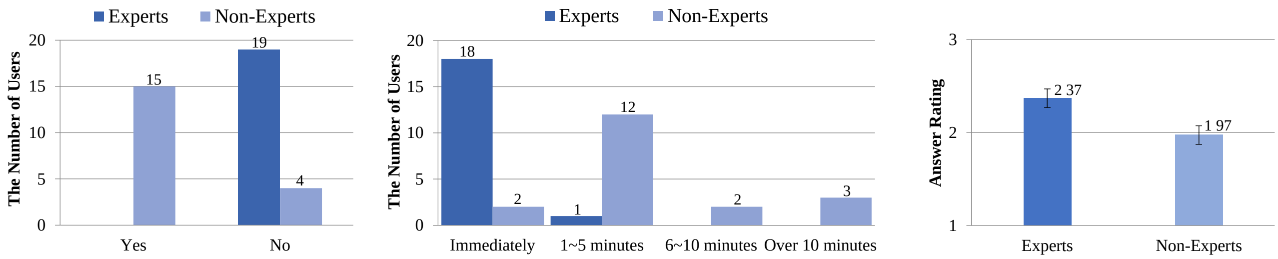
Acknowledgments

This research was supported by Samsung Electronics and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2012R1A1A1012954).

References

Amin, A.; Townsend, S.; van Ossenbruggen, J.; and Hardman, L. 2009. Fancy a drink in Canary Wharf?: A user study on location-based mobile search. In *Proc. of the 12th IFIP TC 13 Int'l Conf. on Human-Computer Interaction*, 736–749.

Arrington, M. 2008. ChaCha and Aardvark: Putting humans to work to get you the answers you need. http://techcrunch.com/2008/11/03/. Accessed: 03-12-2013.



(a) External information. (b) Response time. (c) Answer quality.

Figure 6: The results of the user study.

Bulut, M. F.; Yilmaz, Y. S.; and Demirbas, M. 2011. Crowdsourcing location-based queries. In *Proc. of the 2nd IEEE Workshop on Pervasive Collaboration and Social Networking*, 513–518.

Church, K.; Neumann, J.; Cherubini, M.; and Oliver, N. 2010. SocialSearchBrowser: a novel mobile search and information discovery tool. In *Proc. of the 2010 Int'l Conf. on Intelligent User Interfaces*, 101–110.

Cleave, E. 2012. Google and Bing—desktop search surge in October. http://www.zenergyworks.com/tag/desktop-search/. Accessed: 03-12-2013.

Dice, L. 1945. Measures of the amount of ecologic association between species. *Ecology* 26(3):297–302.

Evans, B. M., and Chi, E. H. 2008. Towards a model of understanding social search. In *Proc. of the 2008 ACM Conf. on Computer Supported Cooperative Work*, 485–494.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *Proc. of the 22nd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, 50–57.

Horowitz, D., and Kamvar, S. 2010. The anatomy of a large-scale social search engine. In *Proc. of the 19th Int'l Conf. on World Wide Web*, 431–440.

Kamvar, M., and Baluja, S. 2006. A large scale study of wireless search behavior: Google mobile search. In *Proc. of the 2006 Conf. on Human Factors in Computing Systems*, 701–709.

Li, B., and King, I. 2010. Routing questions to appropriate answerers in community question answering services. In *Proc. of the 19th ACM Int'l Conf. on Information and Knowledge Management*, 1585–1588.

Mihalcea, R.; Corley, C.; and Strapparava, C. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proc. of the 21st National Conf. on Artificial Intelligence*, 775–780.

Poulin, R. 2003. The decay of similarity with geographical distance in parasite communities of vertebrate hosts. *Journal of Biogeography* 30(10):1609–1615.

Richardson, M., and White, R. W. 2011. Supporting synchronous social Q&A throughout the question lifecycle. In *Proc. of the 20th Int'l Conf. on World Wide Web*, 755–764.

Shankar, P.; Huang, Y.-W.; Castro, P.; Nath, B.; and Iftode, L. 2012. Crowds replace experts: Building better location-based services using mobile social network interactions. In *Proc. of 2012 IEEE Int'l Conf. on Pervasive Computing and Communications*, 20–29.

Si, X.; Chang, E. Y.; Gyöngyi, Z.; and Sun, M. 2010. Confucius and its intelligent disciples: Integrating social with search. *Proc. of the VLDB Endowment* 3(1-2):1505–1516.

Song, Y.; Ma, H.; Wang, H.; and Wang, K. 2013. Exploring and exploiting user search behavior on mobile and tablet devices to improve search relevance. In *Proc. of the 22nd Int'l Conf. on World Wide Web*, 1201–1212.

Steiber, R. 2012. Advertising, mobile-first. *Presented at the panel session of 2012 Open Mobile Summit*.

Tan, J.; Gao, H.; Hu, X.; and Liu, H. 2013. Exploiting homophily effect for trust prediction. In *Proc. of the 6th ACM Int'l Conf. on Web Search and Data Mining*, 53–62.

Wu, Z., and Palmer, M. 1994. Verbs semantics and lexical selection. In *Proc. of the 32nd Annual Meeting on Association for Computational Linguistics*, 133–138.

Yi, J.; Maghoul, F.; and Pedersen, J. 2008. Deciphering mobile search patterns: A study of yahoo! mobile search queries. In *Proc. of the 17th Int'l Conf. on World Wide Web*, 257–266.

Zhou, Y.; Cong, G.; Cui, B.; Jensen, C. S.; and Yao, J. 2009. Routing questions to the right users in online communities. In *Proc. of the 25th Int'l Conf. on Data Engineering*, 700–711.

A Topic Category Weight

We looked into more than 1,100 location-based queries collected from a mobile Q&A service *Knowledge Log*[6]. Those queries were found by searching for venue names and region names on the service. Each of them was manually marked by its intention (or goal) as one of direction, price, service, and realtime. Then, we manually determined which topic categories occurred in each query. Last, the frequency of each topic category was determined by Table 7. Glaucus uses the *percentage* value of the corresponding topic category as w_{cat}.

Table 7: The weight of each topic category.

Intention	Topic Category	Count	Percentage(%)
Direction	L	689	53.7
	C	142	11.1
	T	-	0.0
	N	452	35.2
Price	L	7	4.8
	C	2	1.4
	T	21	14.5
	N	115	79.3
Service	L	14	7.7
	C	1	0.6
	T	10	5.5
	N	156	86.2
Realtime	L	95	62.1
	C	9	5.9
	T	-	0.0
	N	49	32.0

[6]http://www.jisiklog.com/