Spatial Keyword Querying

Xin Cao[1], Lisi Chen[1], Gao Cong[1], Christian S. Jensen[2], Qiang Qu[2], Anders Skovsgaard[2], Dingming Wu[3], and Man Lung Yiu[4]

[1] Nanyang Technological University
[2] Aarhus University
[3] Hong Kong Baptist University
[4] Hong Kong Polytechnic University

Abstract. The web is increasingly being used by mobile users. In addition, it is increasingly becoming possible to accurately geo-position mobile users and web content. This development gives prominence to spatial web data management. Specifically, a spatial keyword query takes a user location and user-supplied keywords as arguments and returns web objects that are spatially and textually relevant to these arguments. This paper reviews recent results by the authors that aim to achieve spatial keyword querying functionality that is easy to use, relevant to users, and can be supported efficiently. The paper covers different kinds of functionality as well as the ideas underlying their definition.

1 Introduction

The sales of Internet-worked mobile devices such as smartphones are skyrocketing. According to the market research firm IDC, some 490 million smartphones were sold worldwide in 2011, a year-on-year increase of nearly 60%. During the first quarter of 2012, worldwide smartphone sales were 145 million, and expectations for 2012 in the range from about 650 to about 690 million units have been reported. Sales are expected to reach 1 billion by 2016, which will be about half of the mobile device market [12].

As a result of this development, the web is accessed increasingly by mobile users. In recognition of this development, Google announced in 2010 that the company would now develop services "mobile first," meaning that services are developed first for mobile devices and users and only then adapted to desktop devices and users.

Next, we are witnessing a proliferation of geo-positioning capabilities. Smartphones, navigation devices, some tablets, and other mobile devices are equipped with GPS receivers. The Galileo global satellite navigation system, which will provide more and better services than currently offered by the GPS system, is scheduled to become operational by the end of the decade.

Other available positioning technologies exploit the communication infrastructures used by mobile devices, e.g., Wi-Fi, 3G, and 2G. Stated briefly, this can be achieved by building and maintaining a so-called location fingerprint database of pairs of a ground truth location and the base stations and cell towers seen by

P. Atzeni, D. Cheung, and R. Sudha (Eds.): ER 2012, LNCS 7532, pp. 16–29, 2012.

the radios in a mobile device from that location. It is then possible to assign a location to a device when the device reports the base stations and cell towers seen from its location. This type of technology may be used for both outdoor and indoor positioning, but offers less accurate positioning than does GPS.

As a result of the developments outlined above, a spatial, or geographical, web is emerging where content and users are associated with locations that are used in a wide range of location-based services. Billions of queries are being processed by web search engines. According to one report, Google processed a daily average of 4.7 billion queries in 2011 [20]. A substantial fraction of these have local intent and target so-called spatial web objects or places, i.e., points of interest with a web presence that have locations as well as textual descriptions. One article reports that 53% of mobile searches on Bing have local intent [21]. An older, PC-centric finding from Google is that 20% of Google searches are related to location [10].

A prototypical spatial keyword query takes a user location and user-supplied keywords as arguments and returns objects that are spatially and textually relevant to these arguments. Due perhaps to the rich semantics of geographical space and the importance of geographical space to our daily lives, many different kinds of relevant spatial keyword query functionality may be envisioned. A range of contributions are already available in the literature that study different aspects of spatio-textual querying (e.g., [5, 6, 8, 14, 16, 18, 25–27]).

This paper discusses contributions by the authors that aim to improve the support currently available for web querying with local intent. In doing so, it describes the general setting of spatial web querying. It surveys fundamental spatial web querying functionality as well as examples of advanced functionality. The objective is to provide the reader with an overview so that interested readers may follow pointers to additional, in-depth coverage of the different functionalities covered. The paper purposefully does not cover implementation and performance aspects, but rather adopts a focus on semantics and aims to explain the ideas and rationale underlying the functionality it covers.

Section 2 presents the setting, covering the modeling of users and content, the modeling of the space in which users and content are embedded, notions of spatial distance and text similarity, and spatial web object ranking functions. Section 3 covers standard and moving spatial keyword querying functionality. It then covers techniques that aim to capture and exploit the benefits of co-location to compute better query results and techniques that allow users to retrieve sets of spatial web objects that jointly satisfy a query. Finally, Section 4 discusses some of the many challenges yet to be addressed in full.

2 Problem Setting

2.1 Content and Users

The formalization of the problem setting is simple. The content that is queried is a set of objects $\mathcal{D}$ that are available on the web and that we often call *spatial*

web objects or *places*. A place $p \in \mathcal{D}$ has two attributes: $\langle \lambda, \psi \rangle$, where λ encodes an accurate geo-location and ψ is a text value.

Increasing volumes of places, i.e., pairs of a location and a text document, are available on the web. Websites for businesses, e.g., bars, cafes, restaurants, banks, dentists, doctors, pharmacies, tourist attractions, hotels, shops, public offices, typically list street addresses. And more and more businesses are acquiring a web presence. Further, business directories list places with descriptions that include location and text, in addition to, e.g., photos, phone numbers, and opening hours. Indeed, there are good reasons for businesses to appear in business directories. A prominent one is that they may then occur more prominently in web query results.

In addition, research is being conducted that aims to associate meaningful locations with web pages based on analyses of the text content on the web pages. This will further increase the volumes of places.

Another type of content that conforms to the assumed format is geocoded microposts, e.g., tweets posted on Twitter. While a micropost carries different semantics than what we associate with a place, spatial keyword querying can also be applied to such content.

We assume that users also have known geo-locations. We also generally assume that the users are mobile. Thus, typical users include individuals with smartphones, tablets, laptops, or similar mobile devices. In the work covered here, we assume that the users have accurate geo-locations, as can be provided by GPS. However, in other settings, it may be assumed that the location of a user is inaccurate and is known only at the granularity of a country, city, or city block. For example, the Google latitude API supports "accurate" and city-level granularities [9]. Or it may be approximated as the region associated with an internet service provider or the service region of a communication cell tower.

2.2 Spaces, Positions, and Distances

The geo-location of a place is modeled as a point location in the space that is used for the modeling of geographical space. We typically assume that geographical space is modeled as two-dimensional Euclidean space. We then typically use Euclidean distance, denoted as $\|\cdot\ \cdot\|$, as the distance notion. We may also use squared Euclidean distance in cases where only relative distance is important because it is simpler than Euclidean distance. These modeling choices are helpful because they are simple and allow us to abstract away the intricacies of how to actually capture geo-locations.

However, we may note that it is also possible to model space as some form of spatial network [24]. Spatial networks are often used for the modeling of transportation networks such as road networks. They are relevant because it is often reasonable to view movement as being constrained to a transportation network and because places are often reachable via such a network.

Spatial network models come in different variations. The simplest may be that of a regular undirected or directed graph where each vertex has a Euclidean point location and where each edge has a weight that captures the edge's lengths.

However, much more accurate spatial network models exist. For example, models may associate polylines, possibly at different levels of detail [11], with the edges to capture the embedding of roads into geographical space. In addition, a model may capture traffic lanes and traffic regulations [19].

In a spatial network, the position of an object is often given by an edge and a distance from the start of the edge, or it is simply assumed that objects can only be located at vertices, in which case a position is given by a vertex.

In the context of spatial network models, the relevant distance notion is that of spatial network distance. The distance from one object to another is then the length of the shortest possible path from the source object to the target object. In the context of spatial networks, travel time may also be considered as the relevant distance notion. In that case, travel times, sometimes time varying, are associated with edges. It may also be relevant to use a notion distance that is based on the assignment of eco weights to edges, thus capturing the greenhouse gas emissions of vehicles that traverse the edges.

Substantial infrastructure is available for associating places with locations and vice-versa. Internet Yellow Pages and online business directories (e.g., Bing Business Portal, Google Places, and Yahoo! Local) offer listings that associate businesses with street addresses. A geocoder is a service that typically maps a street address to a geo-location, often a latitude-longitude pair (corresponding to a two-dimensional Euclidean point location). A reverse geocoder is then the opposite: it maps a geo-location to a street address.

Finally, we observe that it is becoming increasingly relevant to also consider indoor spaces [13]. In that case, locations may be symbolic, e.g., room names or numbers, and indoor walking distance may be used as the distance notion [15].

2.3 Text Content and Relevance

The second attribute of a place is a text value or a document. We generally assume that this value is a simple, unstructured text string. As we shall see in detail later, queries also take arguments that are (shorter) text strings and that we call query keywords. Just like we need a means of quantifying the distance between two positions, we need a means of comparing the relevance of one text value (of a place) to another text value (typically of a query). To this end, we can use one of several state-of-the-art techniques.

Specifically, with p being a place and $q = \langle \lambda, \psi \rangle$ being a prototypical query that takes both a location and a text value as arguments, we let the function $tr_{q.\psi}(p.\psi)$ denote the text relevancy of $p.\psi$ to $q.\psi$. The text relevance $tr_{q.\psi}(p.\psi)$ can be computed using an information retrieval model, such as a language-based model [7] or the vector space model [3], which we review next.

The text relevance between term t and $p.\psi$ can be computed by a language model [17] as:

$$\hat{p}(t|\theta_{p.\psi}) = (1-\lambda)\frac{tf(t,p.\psi)}{|p.\psi|} + \lambda\frac{tf(t,Coll)}{|Coll|} \quad (1)$$

Here, $tf(t,d)$ takes a term t and a document d as arguments and returns the number of occurrences of the t in d; $Coll$ is the document that consists of

the collection of all documents associated with places in $\mathcal{D}$; $tf(t,d)/|d|$ is the maximum likelihood estimate of t in d; and λ is a smoothing parameter of the Jelinek-Mercer smoothing method. Finally, text relevance for the language model approach is computed as follows:

$$tr^{1}_{q.\psi}(p.\psi) = \prod_{t \in q.\psi} \hat{p}(t|\theta_{p.\psi}) \tag{2}$$

The text relevance can also be computed by the vector space model [28] as follows:

$$\begin{aligned} &tr^{\text{vs}}_{q.\psi}(p.\psi) = \frac{\sum_{t \in q.\psi \cap p.\psi} w_{q.\psi,t} w_{p.\psi,t}}{W_{q.\psi} W_{p.\psi}}, \text{ where } w_{q.\psi,t} = \ln(1 + \frac{|Coll|}{f_t}), \\ &w_{p.\psi,t} = 1 + \ln(tf(t, p.\psi)),\ W_{q.\psi} = \sqrt{\sum_t w^2_{q.\psi,t}},\ W_{p.\psi} = \sqrt{\sum_t w^2_{p.\psi,t}} \end{aligned} \tag{3}$$

Here f_t is the number of objects whose text descriptions contain the term t, and $tf(t, p.\psi)$ is the frequency of term t in $p.\psi$; $w_{p.\psi,t}$ corresponds TF and $w_{q.\psi,t}$ corresponds to IDF.

Finally, it may be observed that much text content available on the web has structure that may be exploited for computing text similarity. This then calls for new definitions of relevance.

2.4 Place Ranking

So far, we have seen that places have both a geo-location and a textual description, and we have discussed means of comparing locations with locations and textual descriptions with textual descriptions. The prototypical query $q = \langle \lambda, \psi \rangle$ takes both a location and a text argument and retrieves places that are in some sense relevant to both arguments. As a foundation for defining a variety of queries, we then need a means of determining how well a place matches a query. Intuitively, a place whose text description is more relevant to the query's text argument and whose location is closer to the query location is preferable.

In our work, we have considered two general ways of accomplishing this. First, we can rank a place with respect to a query as follows.

$$rank^{1}_{q}(p) = \alpha \frac{\|q.\lambda\ p.\lambda\|}{maxD} + (1 - \alpha) \frac{tr_{q.\psi}(p.\psi)}{maxP} \tag{4}$$

This definition builds on and combines the earlier definitions of distance and text relevance. It normalizes each by the maximum possible distance and relevance so that each becomes a value between 0 and 1. It also introduces a parameter α that makes it possible to balance the importance of the two aspects.

Second, we may use a function that weights the geographical distance by the text relevance [1].

$$rank^{2}_{q}(p) = \frac{\|q.\lambda\ p.\lambda\|}{tr_{q.\psi}(p.\psi)}, \tag{5}$$

With this function, the smaller a value of $rank_q(p)$ a place p gets, the more relevant it is to query q, and the higher it is ranked.

When setting α to 0 in Definition 4, the definition degenerates to textual relevance as used by a search engine; and setting α to 1, the definition considers only distance and degenerates to what is used in a standard nearest neighbor query. On the other hand, we are also left with the problem of providing an appropriate value for α—it is unlikely to be a good idea to ask users to set α. Definition 5 has the advantage that no parameter needs to be set, but then it of course also does not offer the control provided by the parameter.

The two definitions of ranking take into account two aspects, or signals. A commercial web search engine is likely to take into account many more signals, PageRank being a well-known one. In our setting, directions queries (users who ask for directions from a place a to a place b) can be used for the ranking of places [22]. The general idea is that a query for directions to place b is a vote that place b is interesting. As an example of further aspects of the problem, if the distance from place a to place b is long, this may be taken as an indication that place b is more interesting than if the distance is short. Another type of user-generated content, namely GPS traces, can also be used for the ranking of places [2]. It is easy to envision how additional signals can be accounted for by the introduction of additional terms in Definition 4.

3 Spatial Keyword Queries

3.1 Standard Queries

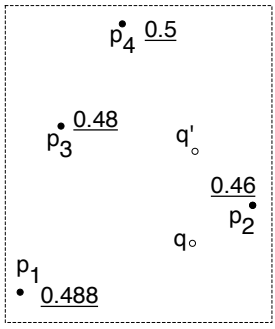
Based on the setting provided in the previous section, we proceed to describe what may be considered as standard spatial keyword queries. These all involve different conditions on the spatial and textual aspects of places. In spatial databases, the arguably most fundamental queries are range queries and k nearest neighbor queries. In text retrieval, queries may be Boolean, requiring results to contain the query keywords, or ranking-based, returning the k places that rank the highest according to a text similarity function as discussed in Section 2.3.

The *Boolean range query* $q = \langle \rho, \psi \rangle$, where ρ is a spatial region and ψ is a set of keywords, returns all places that are located in region ρ and that contain all the keywords in ψ. Variations of this query may rank the qualifying places. The *Boolean kNN query* $q = \langle \lambda, \psi, k \rangle$ takes three arguments, where λ is a point location, ψ is as above, and k is the number of places to return. The result consists of up to k places, each of which contain all the keywords in ψ, ranked in increasing spatial distance from λ.

Next, the *top-k range query* $q = \langle \rho, \psi, k \rangle$, where ρ, ψ, and k are as above, returns up to k places that are located in the query region ρ, now ranked according to their text relevance to ψ. Finally, the *top-k kNN query* takes the same arguments as the Boolean kNN query. It retrieves k objects ranked according to a score that takes into consideration spatial proximity and text relevance, as discussed in Section 2.4.

Among these queries, the latter two ones that perform textual ranking are the most similar to standard web querying, and the last one is the one that is most interesting and novel, as it integrates the spatial and textual aspects in the ranking. This query, also called the *top-k spatial keyword query*, is exemplified in Figure 1.

Figure 1(a) shows the locations of a set of objects $\mathcal{D} = \{p_1, p_2, p_3, p_4\}$. Assume that we are given the query q shown in Figure 1(a) with $q.\psi = \langle a, b \rangle$ and $q.k = 2$. The underlined number next to each object is its text relevancy to the query keywords $q.\psi$. These are computed using a text relevancy function $tr_{q.\psi}(p.\psi)$. The result of query q is $\langle p_2, p_3 \rangle$ according to function $rank_q^2(\cdot)$. The ranking values of p_2 and p_3 are 0.478 (= 0.22/0.46) and 0.54 (= 0.26/0.48), respectively.



(a) Locations

	$\|q\ p_i\|$	$\|q'\ p_i\|$
p_1	0.28	0.30
p_2	0.22	0.22
p_3	0.26	0.24
p_4	0.28	0.24

(b) Distances

Fig. 1. Top-k Spatial Keyword Query

3.2 Moving Queries

The queries considered above are one-time queries: a query is issued and the result is computed and returned to the user; end of story. This is in contrast to a continuous query that is a one-time query that is registered with the system and is later deregistered. As long as the query remains registered, the system must send a new result to the user each time the result changes. In our setting, contributions have so far been assumed that set $\mathcal{D}$ is static, so no changes occur.

A moving query is akin to continuous queries. Specifically, a *moving top-k spatial keyword query* takes the normal arguments ψ and k, but then it also takes as argument a location λ that changes continuously and is intended to capture the user's location [23]. The system must now send the user a new result every time the result changes due to changes in λ.

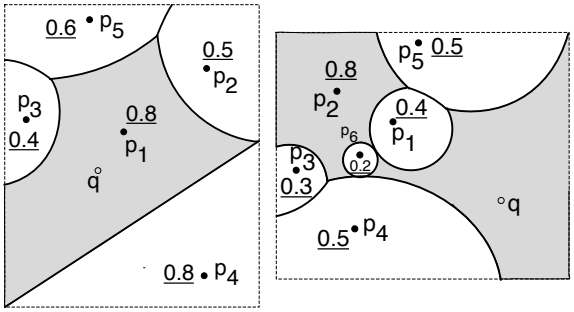
The state-of-the-art approach to handling such queries is to utilize safe zones. When the result of a query is computed, the system also computes a safe zone, which is a region with the property that as long as λ remains inside the region, the result is guaranteed not to change. This region is sent to the user (to the part of the system on the user's mobile device) along with the result. Now the client side can monitor λ and notify the server side when λ is about to exit the safe zone, upon which the story repeats itself.

Returning to the example in Figure 1, when q moves to q', the result becomes $\langle p_2, p_4 \rangle$. The ranking value of p_2 and p_4 is 0.478 and 0.48, respectively.

Readers familiar with spatial databases may know that the safe zone for a standard nearest neighbor query is a cell in a Voronoi diagram. It turns out that if we adopt the ranking function in Equation 5, the safe zone for our problem is

a cell in a k^{th} order multiplicatively weighted Voronoi (mwV) diagram. In our setting, a place that is highly relevant to a query may belong to the query result even if it is located relatively far from the query. Thus, the safe zone for a query result consisting of such a place is a large region. In addition, the safe zone may have "holes" in it that are caused by the presence of places that are less relevant but nearer to the query.

Figure 2(a) shows an mvV diagram and a safe zone for a query q with $k = 1$. The underlined values next to the places are their text relevancies to the query. The safe zone is the region that both q and the result p_1 belong to. While mwV



(a) Safe Zone (b) Disconnected Safe Zone

Fig. 2. Multiplicatively Weighted Voronoi Diagrams

diagrams fit the problem perfectly, their use also raises questions. For example, Figure 2(b) shows a case where the safe zone of the result of a query q ($k = 1$; i.e., p_2) is disconnected, which means that it is impossible for the user to reach p_2 without another place becomes a better result along the way. So if users issue continuous queries for places with the intention of visiting the places, is seems becoming to explore other definitions of the ranking.

3.3 Accounting for Co-location

We often see that similar businesses locate near each other. Examples include restaurant districts (e.g., China Town, Little Italy), bar districts, shopping streets, markets and bazaars (e.g., farmer's markets, antiques markets), and regions with a concentration of car sales businesses. This phenomenon seems to suggest that businesses benefit from locating near similar businesses. A possible explanation may be that this affords consumers easy access to a larger selection of products and services. For example, if no shoes in a store are attractive to a consumer, the presence of nearby shoe stores is desirable. Thus, we may expect that concentrations of similar businesses attract proportionally more consumers than do isolated stores. Put differently: co-located stores attract more customers to such an extent that this compensates for the increased competition.

The problem is then how to integrate the benefits of co-location into the ranking of places in top-k spatial keyword queries. To solve this problem, we

have developed an approach that exploits a query centric notion of prestige. Given a query, a place is assigned prestige that takes into account the presence of nearby places that are also relevant to the query [3]. Then the prestige of a place is used for the ranking.

To illustrate prestige-based object ranking, consider the query "shoes" at location q in Figure 3 ($k = 1$). Circles represent shops selling shoes or jeans, with the centers representing the locations and the areas representing the relevancies to the query. Techniques that treat objects independently (e.g., [7]) return p_5

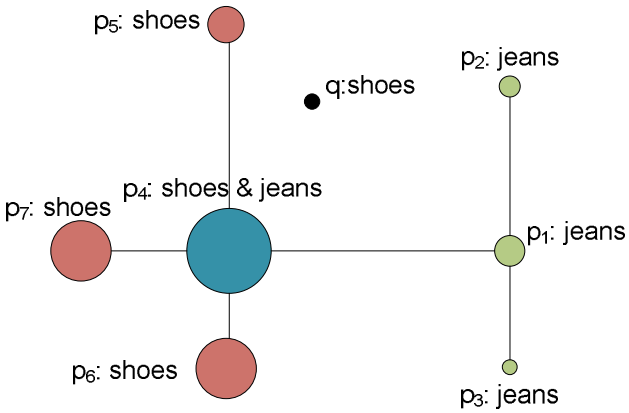


Fig. 3. Prestige Propagation

as the result, since p_5 is relevant and closest to q. However, when using prestige for ranking, p_4 becomes the result because it has more nearby shops that are relevant to the query and also is close to the query location.

We also note that the notion of prestige involves mutual reinforcement: the prestige of one place is affected by the prestige of nearby places. Therefore, we employ a PageRank-like random walk mechanism for the propagation of prestige. The graph that is used is created by connecting places with edges if the places are sufficiently close and similar.

The use of prestige also has the benefit that even if the text of a place does not contain any of the query terms, the place can still be identified as relevant. This occurs if the place has a description that matches those of nearby objects that do contain the query terms. See the places in the figure with text "jeans."

3.4 Beyond the Single-Object Result Granularity

A trend in general web querying is to produce more interesting and richer results than simply lists of blue links with search snippets. For example, a query on google.com for "Caroline Wozniacki" returns not only links, but also information on the performance of tennis star Caroline Wozniacki in the most recent tournament she participated in, a photo of her, and basic information about her such as birth date and place. It also lists names and photos of other tennis stars that people (who search for Caroline Wozniacki) also search for.

Our setting offers new opportunities for departing from the blue links where the granularity of a result is a single link or place. First, queries exist where no single, nearby place is a good result by itself, but where several places near each other in combination constitute a good result. Put differently, the places *collectively* meet the need expressed in the query, while none of the individual places meet need. For example, a "hotel, gym, theater" query may have its best result a pair of a theater and a hotel with a gym that are located next to each other and are close to the query.

Next, note that in the above example, a user would be happy with a hotel with both a gym and theater. In other cases, the user is directly interested in retrieving collections of places that are near each other. Specifically, the user may want to find a *region* of a specified size such that the places in the region meet the user needs in a better way than individual objects. For example, users would prefer to do shopping in a region with many shops.

We proceed to consider spatial keyword querying that targets the above two situations.

Collective Queries. To address the first situation, we consider a query q that takes a user location λ and a set of keywords ψ as arguments. Its search space is all subsets of the set of places $\mathcal{D}$. It returns a set of places such that (i) the textual descriptions of the objects collectively cover ψ, (ii) the result places are all close to λ, and (iii) the result places are close to each other.

These three conditions represent three different aspects of meeting the user's needs, and different instantiations of each condition may be preferable in different application scenarios. We have studied two instantiations of the query [4].

1. Find a group of objects χ that cover the keywords in q such that the sum of their spatial distances to the query is minimized.
2. Find a group of objects χ that cover the keywords in q such that the sum of the maximum distance between an object in χ and query q and the maximum distance between two objects in χ is minimized.

The first instantiation focuses on the distance between the query and each result object. For example, this may fit a scenario where the user is at a hotel and wants to visit nearby places that offer different services (e.g., workout in a gym, get a haircut, watch a movie) but wishes to return to the hotel in-between services. The second instantiation considers all three aspects. This may fit a scenario where the user wishes to visit all the places before returning.

Other types of instantiations exist. For example, we can relax the keyword matching requirement to allow partial matching, and we can consider other distance measures (e.g., the sum of distances) to replace the maximum distance measure. Note also that the collective query reduces to a standard spatial keyword query when individual objects can cover all the query keywords.

Region Queries. Next, we consider the situations where the user is explicitly interested in retrieving a collection of places.

Consider an example where a user wants to buy "blue jeans and fashion shoes" and wants to explore the offerings of different stores that are within walking distance. In this example, the user is interested in a region that is densely populated with stores that sell "blue jeans and fashion shoes." The following are important aspect of the query formulation: (i) the size and shape of the region, (ii) the relevance of the stores in the region to the query, (iii) the number of relevant stores in the region, and (iv) the density of the relevant stores in the region.

We call this a *relevant region query*. We instantiate the query by providing precise definitions for the four aspects. An instantiation should be well defined, should be useful from a user perspective, and should be computationally tractable.

With this in mind, we might consider square regions of fixed size (i.e., both the size and shape of returned regions are fixed). For the second aspect, we can use any information retrieval model for computing relevance scores, and with such scores available, it is straightforward to provide a definition for the third aspect. Finally, to define density, one option is to use the ratio of the area of the region to the number of relevant objects in the region, while another option is to use the average pairwise distance between relevant objects in the region.

Finally, note that the relevant region query reduces to a standard spatial keyword query when the specified region size is sufficiently small.

4 Outlook

In spatial keyword querying, Google-style keyword-based querying is enhanced to offer better support for queries with local intent by exploiting accurate locations associated with the users and content that their queries target. Here, we have surveyed recent results obtained by the authors. We hope that the coverage convinces the reader that the general research topic is interesting and that it may be possible to obtain results in this area that can have significant impact.

Research on spatial web querying has just begun, and there are many opportunities for continued research. Some were already suggested. Here, we proceed to discuss three opportunities and then three challenges that may also help direct future research efforts.

Structured Queries. While we believe that a major reason for the success of Google-style web querying is the simplicity of the interface and the ease and speed with which one can obtain useful results, it may be of interest to study ways of adding structure to spatial web querying.

For example, queries may include a transportation mode attribute that can be set to, e.g., walking, bicycling, public transport, or driving. In many cases, it may be possible to detect the value of this attribute so that it needs not be set explicitly by the user. With such an attribute, it may be possible to deliver more relevant results in a number of use cases.

Next, a scope or region-of-interest may be included that can contribute to improving query results. Opportunities also exist for inferring a setting for this

attribute without involving the user: the relevant scope of a query may be inferred from the user's zoom level, or it may be inferred from the query itself. A query for a convenience store is likely to have a smaller than a query for, say, Ikea.

Amazon-Style and Social Queries. Quite generally, there is a great potential for computing very personalized results. Using the functionality of the Amazon.com website as inspiration, themes such as those illustrated next deserve further study:

Using personal history and the current location: given the places you have visited, you may be interested in the following nearby places: ...
Using the histories of others at the current location: people who were at your location visited ...
Using the histories of similar others and the current location: when people, who visited places that you have visited were at your current location, they visited ...

It is also generally true that the recommendations and behaviors of individuals we know carry more significance to us than do those of strangers. Thus, there is great potential for integrating social network functionality into the above.

Feedback Mechanisms. According to Tim O'Reilly, "Figuring out how to build databases that get better the more people use them is the secret source of every web 2.0 company." This is a very important insight. In addition, services should learn and adapt to the preferences of their users. Indeed, web search engines rank results according to how people link from one page to another. And folklore has it that search engines observe which links in query results the users click. This offers a mechanism for improving. For example, if users frequently click the fifth link in the result of a particular query, this suggests that the fifth link should be placed higher in the result of that query.

Similarly, we need mechanisms that observe user actions in response to the results they receive. For example, we could observe the actual movement behaviors of users and relate these to the results we provided to the users.

Avoiding Parameter Overload. As a research topic matures and the most obvious functionality, we often see a move towards exploring more complex functionality that calls for the introduction of problem parameters.

The top-k spatial keyword query (Section 3.1) took three parameters: λ (location), ψ (keywords), and k (result cardinality), with only λ being new in comparison to standard keyword querying. Because it can be assumed that λ can be obtained automatically, this query is just as simple to formulate as a standard query. But then we introduced a parameter α for balancing the influence of spatial proximity and text relevance in the first of the two ranking functions for spatial keyword queries that we covered (Equation 4). And it is easy to imagine a more sophisticated ranking function that takes into consideration additional aspects (Recall, e.g., the relevant region query in Section 3.4.).

But then additional parameters may also be needed to control and balance the aspects.

The introduction of parameters that are exposed to the users or that are simply hard to set comes at a cost. Stated very generally, the relevance of a proposal often decreases rapidly as the number of parameters increases.

User Evaluation. A good first step when trying to improve something is to be able to reliably measure that something.

When we develop, say, a new join algorithm, the result is well defined. So in an empirical study of such an algorithm, we simply need to verify that the implementation computes the right result and can then focus on aspects such as the runtime or I/O performance that are fairly easy to measure. Furthermore, empirical studies can often be done meaningfully with synthetic data.

In the context of spatial keyword querying, things are more difficult. First, we typically do not merely aim for a correct result, but rather aim for a result that is as useful as possible for an important use case. But utility depends on the users. Thus, we need to determine how useful users will find a result. It is a complex challenge to establish a reliable ground truth for the result of a query. This is also a challenge that systems-oriented database researchers are not well equipped to address. And it is a challenge that it is costly to address. We feel that much can be gained from working with more user-focused scientists on evaluations (in addition to other aspects). Second, we observe that many problems within this research topic are difficult to address meaningfully without real data. In sum, when considering a new problem, two of the first questions to ask are how the results are to be evaluated and whether real data is available.

Tractability Versus Utility. On the one hand, we aim to support the functionality that (we think) users want. On the other hand, we need to ensure that the functionality that we provide can be supported efficiently. In spatial keyword querying, this tradeoff between tractability and utility of functionality is particularly prominent. For example, queries that return sets of places (recall Section 3.4) may benefit from considering all subsets of places and thus easily become NP hard. It is an interesting challenge to manage this tradeoff.

References

1. Aurenhammer, F., Edelsbrunner, H.: An optimal algorithm for constructing the weighted Voronoi diagram in the plane. Pattern Recognition 17(2), 51–57 (1984)
2. Cao, X., Cong, G., Jensen, C.S.: Mining significant semantic locations from GPS data. PVLDB 3(1), 1009–1020 (2010)
3. Cao, X., Cong, G., Jensen, C.S.: Retrieving top-k prestige-based relevant spatial web objects. PVLDB 3(1), 373–384 (2010)
4. Cao, X., Cong, G., Jensen, C.S., Ooi, B.C.: Collective spatial keyword querying. In: SIGMOD, pp. 373–384 (2011)
5. Chen, Y.-Y., Suel, T., Markowetz, A.: Efficient query processing in geographic web search engines. In: SIGMOD, pp. 277–288 (2006)

6. Christoforaki, M., He, J., Dimopoulos, C., Markowetz, A., Suel, T.: Text vs. space: efficient geo-search query processing. In: CIKM, pp. 423–432 (2011)
7. Cong, G., Jensen, C.S., Wu, D.: Efficient retrieval of the top-k most relevant spatial web objects. PVLDB 2(1), 337–348 (2009)
8. Felipe, I.D., Hristidis, V., Rishe, N.: Keyword search on spatial databases. In: ICDE, pp. 656–665 (2008)
9. Google: Google latitude API (2012), http://developers.google.com/latitude
10. Google: Google Places. Stats & Facts (2012), sites.google.com/a/pressatgoogle.com/googleplaces/metrics
11. Hage, C., Jensen, C.S., Pedersen, T.B., Speicys, L., Timko, I.: Integrated data management for mobile services in the real world. In: VLDB, pp. 1019–1030 (2003)
12. IDC: Smartphone statistics and market share (2012), www.email-marketing-reports.com/wireless-mobile/smartphone-statistics.htm
13. Jensen, C.S., Lu, H., Yang, B.: Graph model based indoor tracking. In: MDM, pp. 122–131 (2009)
14. Li, Z., Lee, K.C.K., Zheng, B., Lee, W.-C., Lee, D.L., Wang, X.: IR-tree: an efficient index for geographic document search. TKDE 23(4), 585–599 (2011)
15. Lu, H., Cao, X., Jensen, C.S.: A foundation for efficient indoor distance-aware query processing. In: ICDE, pp. 438–449 (2012)
16. Lu, J., Lu, Y., Cong, G.: Reverse spatial and textual k nearest neighbor search. In: SIGMOD, pp. 349–360 (2011)
17. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: SIGIR, pp. 275–281 (1998)
18. Rocha-Junior, J.B., Gkorgkas, O., Jonassen, S., Nørvåg, K.: Efficient Processing of Top-k Spatial Keyword Queries. In: Pfoser, D., Tao, Y., Mouratidis, K., Nascimento, M.A., Mokbel, M., Shekhar, S., Huang, Y. (eds.) SSTD 2011. LNCS, vol. 6849, pp. 205–222. Springer, Heidelberg (2011)
19. Speicys, L., Jensen, C.S.: Enabling location-based services—multi-graph representation of transportation networks. Geoinformatica 12(2), 219–253 (2008)
20. Statistic Brain: Google Annual Search Statistics (2012), http://www.statisticbrain.com/google-searches/
21. Search Engine Land: Microsoft: 53 percent of mobile searches have local intent (2012), searchengineland.com/microsoft-53-percent-of-mobile-searches-have-local-intent-55556
22. Venetis, P., Gonzalez, H., Jensen, C.S., Halevy, A.: Hyper-local, directions-based ranking of places. PVLDB 4(5), 290–301 (2011)
23. Wu, D., Yiu, M.L., Jensen, C.S., Cong, G.: Efficient continuously moving top-k spatial keyword query processing. In: ICDE, pp. 541–552 (2011)
24. Zeiler, M.: Modeling our World—The ESRI Guide to Geodatabase Design. ESRI Press (1999)
25. Zhang, D., Chee, Y.M., Mondal, A., Tung, A.K.H., Kitsuregawa, M.: Keyword search in spatial databases: Towards searching by document. In: ICDE, pp. 688–699 (2009)
26. Zhang, D., Ooi, B.C., Tung, A.K.H.: Locating mapped resources in web 2.0. In: ICDE, pp. 521–532 (2010)
27. Zhou, Y., Xie, X., Wang, C., Gong, Y., Ma, W.-Y.: Hybrid index structures for location-based web search. In: CIKM, pp. 155–162 (2005)
28. Zobel, J., Moffat, A.: Inverted files for text search engines. ACM Comput. Surv. 38(2) (2006)