smartphone, or the like. A computer is a device having one or more general or special purpose processors, memory, storage, and networking components (either wired or wireless). The device executes an operating system, for example, a Microsoft Windows-compatible operating system (OS), Apple OS X or iOS, a Linux distribution, or Google's Android OS. In some embodiments, the client device 101 may use a web browser 113, such as Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari and/or Opera, as an interface to interact with the online booking system 111. In other embodiment, the client device 101 can execute a dedicated application for accessing the online booking system 111.

[0011] The online booking system 111 includes web server 109 that presents web pages or other web content, which form the basic interface to the users 103. Users 103 use respective client devices 101 to access one or more web pages, and provide data to the online booking system 111.

[0012] The online booking system 111 may be for example an accommodation reservation system, a dining reservation system, a rideshare reservation system, a retail system, and the like. More generally, the online booking system 111 provides users with access to an inventory of resources (e.g. goods and services) that are available to consumers, and for which the real world, physical location of the resource is considered as an intangible factor in the consumer's decision to consume (e.g., purchase, rent, or otherwise obtain) the resource. Generally, resources available at some locations are more desirable than otherwise identical resources available at other locations. Resources include accommodations; restaurants; vehicles; attractions (e.g., shows, events, tourists attractions); shopping centers; and the like. For example, in an online booking system 111 that provides accommodations, accommodations in particular neighborhoods may be more or less desirable than otherwise identical accommodations in some neighborhood: a given neighborhood may be considered more interesting, more prestigious, safer, or have some other quality that consumers deem valuable when selecting accommodations.

[0013] In some embodiments, the online booking system 111 facilitates transactions between users 103. For example, an accommodation reservation system allows users 103 to book accommodations provided by other users of the accommodation reservation system. A rideshare reservation system allows users 103 to book rides from one location to another. An online market place system allows users 103 to buy and/or sell goods or services face to face with other users. The online booking system 111 comprises additional components and modules that are described below.

ranking is location relevance score that is based on the real world location of the listing to be ranked and a real world location specified in a received search query. The determination of location relevance score is performed by a location relevance module 213, which is further described below. Other possible factors include, for example, the price of a listing, the number and quality of reviews of the listing provided by other users, the quality of pictures in the list, the number of successful and unsuccessful prior bookings, a reply rate, searching user behavioral signals (like clickthrough from search to listing view or listing view to booking), whether the listing is associated to the searcher via a social networking system social graph.

[0024] The ranking module 211 uses the stored historical search, booking, and listing information in order to rank listings. To facilitate this, the search module 205 and booking modules 207 store search, listing browsing, and booking information in database 201. This historical information is stored on a per user, per web browsing session basis, such that a user's interactions with the online booking system 111 are stored together, including any search queries entered, any listings viewed, and any bookings made. Storing search queries and subsequent bookings together, particularly allows the online booking system 111 to aggregate useful statistics across many different users. For example based on stored historical bookings and historical search queries, the online booking system 111 is capable of determining for a given search query (or portion thereof), what bookings were made by users who entered that search query. The online booking system can also determine the reverse: for a given booking, what search queries were made by users.

[0025] In using historical search, booking, and listing information to rank listings, the ranking module 211 may use any historical period. For example, the ranking module 211 may use listings, bookings, and searches that occurred in the last month, last three months, last 6 months, last year, all time, or any period therebetween. Alternatively, the ranking module 211 may use listings, bookings, and searches from particular periods (e.g., that occurred during a particular season such as winter, or that occurred over Thanksgiving weekend, etc.).

[0026] In situations in which the systems discussed here collect personal information about users, or may make use of personal information, the users may be provided with an opportunity to control whether programs or features collect or store user information (e.g., whether to maintain historical listings, historical search queries, and historical bookings), or to control whether and/or how to receive content from the online booking system 111 that may be more relevant to the user. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be

frequently visit. This city relevance subscore adjusts for this lack of user information about nearby cities by using historical information about which cities users actually booked in for particular search queries. In very broad terms, the more frequently users booked in a given city A, when searching for ~~for~~ city B, the more relevant city A is to a search query for city B. The city relevance subscore R_{Ci} precisely quantifies this relevance relationship.

[0030] The location relevance module 213 determines a city relevance subscore R_{Ci} for one or more cities $*$. For a given listing i in listing city C_i, the module 213 assigns the city relevance subscore R_{Ci} for city C_i to listing i. To determine the city relevance subscore R_{Ci}, the location relevance module 213 uses the query country Q_{I*} and (where applicable) the query state Q_{S*} to uniquely identify the query city Q_{C*}, as many different cities around the world share the same name. For example, Springfield Missouri and Springfield, Illinois share the same city name Springfield but are located in different states. Using the geocoded query country Q_{I*} and in some cases the query state Q_{S*} allows for unique identification of the query city Q_{C*}.

[0031] The location relevance module 213 identifies a set of available listings L within a threshold distance to the query city Q_{C*} (e.g., in kilometers) . This includes listings within the query city Q_{C*}, and may also include listings in other nearby cities, some of which may be in neighboring states and countries.This may include only currently available listings, or it may also include historical listings that are no longer available. In the following description, L will also refer to total number of listings in this set, as will be apparent from the context. The listings L are obtained from database 201.

[0032] The location relevance module 213 subdivides the set of listings L into subsets, one listing subset L_C for each city having listings in the set L. In the following description, L_C will also refer to the total number of listings in each city, as will be apparent from the context. The subset of listings for the query city Q_{C*} is L_{C*}. The number of listings in each city L_C and in the set L is useful for normalizing the city relevance subscore to avoid unduly skewing the city relevance subscore R_{Ci} towards large or small cities.

[0033] The location relevance module 213 further uses the query country Q_{I*}, query state Q_{S*}, and query city Q_{C*} to identify historical search queries from database 201 that share the same Q_I, Q_S, and Q_C. Many of the historical search queries will have resulted in historical bookings that occurred subsequent to the user inputting the search query, often during the same web browsing session. However, as above it is not necessarily the case that those prior users ended up booking listings in the same city as the query cities Q_C of their historical

search queries. For example, many users searching for "Santa Cruz" as their query city, may end up booking in Aptos, which is nearby.

[0034] Using the historical bookings in database 201, the module 213 identifies a total number of bookings $B_Q(C_i)$ in each listing city C_i in which a user ended up booking a listing in after searching using the query city Q_{C*}, query country Q_{I*}, and query state Q_{S*} (where applicable). The bookings from these cities are also aggregated to determine a total number of historical bookings B_{QT}, given the query city Q_{C*}, query country Q_{I*}, and query state Q_{S*} (also where applicable). For example, a search for "Santa Cruz" may include bookings in Santa Cruz, Aptos, Capitola, and Soquel.

[0035] The location relevance module 213 uses the above identified quantities to derive additional quantities. The module 213 identifies the probability $P(B_Q(C_i) \mid B_{QT})$ a user will book a listing in a city C_i of a listing *i*, given that they entered a search query containing the query city Q_{C*}, query country Q_{I*}, and query state Q_{S*} (where applicable). This quantity mirrors historically measured behavior, as it may be a common occurrence for users to book in city A after searching for city B. This probability may also be calculated as a percentage/ratio $B_Q(C_i)/B_{QT}$. This quantity expected to be high for most listings where the query city Q_{C*} matches the listing city C_i. However, it also expected to be non-zero for many other listing cities C_i that do not match the query city Q_{C*}.

[0036] The location relevance module 213 also identifies the probability $P(Bq(C_i)|B(C_i))$ a user will book in a city C_i of a listing *i*, relative to the total number of historical bookings $B(C_i)$ that happened in city C_i. This probability may also be calculated as a percentage/ratio of $Bq(C_i)/B(C_i)$. Similarly, this quantity is expected to be high for most listings where the query city Q_{C*} matches the listing city C, and it also expected to be non-zero for many other listing cities C that do not match the query city Q_{C*}.

[0037] In determining the city relevance subscore R_{Ci}, the second probability $P(Bq(C_i)|B(C_i))$ can be used to balance out the first probability from the previous paragraph $P(B_Q(C_i) \mid B_{QT})$. For example, the first probability $P(B_Q(C_i) \mid B_{QT})$ quantifies the probability people search for one city (e.g., Santa Cruz, New York City), but end up booking in another city (e.g., Aptos, Newark). Despite their differences as cities (e.g., Aptos being smaller and less known to users than Newark), this probability may, in some instances, be comparable for both Aptos and Newark. The second probability $P(Bq(C_i)|B(C_i))$, however, may distinguish between these two types of cities. For example, as Aptos is one of a few small satellite cities surrounding Santa Cruz, it may be the case that the second probability $P(Bq(C_i)|B(C_i))$ indicates that a significant number of people booking in Aptos search for Santa Cruz. More

description, L_C will also refer to total number of listings in this set, as will be apparent from the context. The listings L_C are obtained from database 201. The location relevance module 213 subdivides the set of listings L_C into subsets, one subset of listings L_N for each neighborhood in the query city Q_{C*}. In the following description, L_N will also refer to the total number of listings in each neighborhood, as will be apparent from the context. The location relevance module 213 determines the number of listings in a neighborhood $_{LN}$ by accessing the listings in database 201 to identify the neighborhood in which each listing is located. This information may ~~be~~ have been provided by the listing user. Alternatively, neighborhood information may be accessed or provided by an external source. For example, an exogenously generated database may provide correlations between locations and neighborhoods.

[0045] The location relevance module 213 determines the neighborhood relevance subscore R_{Ni} based on at least one of a number of listings L_N in the neighborhood N_i, a probability $P(B_N (N_i)| B_C)$ of booking a listing B_N in neighborhood N_i relative to the total number of bookings B_C in the city C_i, and a probability $P(L_N(N_i) | L_C(C_i))$ of listing *i* being in neighborhood N_i (given by $L_N(N_i)$) relative to the total number of listings L_C in city C_i.

[0046] The location relevance module 213 determines the probability $P(B_N(N_i) | B_C(C_i))$ by accessing historical bookings from database 201. The module 213 identifies the total number of historical bookings of listings in each neighborhood B_N separately for each neighborhood in a city, as well as the total number of historical bookings of listings in the city B_C across all included neighborhoods. The probability $P(B_N(N_i) | B_C)$ may also be calculated as a percentage/ratio $B_N(N_i)/B_C(C_i)$.

[0047] The location relevance module 213 determines the probability $P(L_N(N_i) | L_C(C_i))$ by accessing currently available and/or historical listings from database 201. The module 213 identifies the total number of listings in each neighborhood L_N in a city, as well as the total number of listings in the city L_C across all included neighborhoods. The probability $P(L_N(N_i) | L_C(C_i))$ may also be calculated as a percentage/ratio $L_N(N_i) / L_C(C_i)$.

[0048] In one embodiment, the location relevance module 213 determines the neighborhood relevance subscore R_{Ni} according to:

$$R_{Ni} = \begin{cases} P(B_N(N_i) | B_C) & \text{if } Q_{C*}=C_i \\ 0 & \text{if } Q_{C*}\neq C_i \end{cases}$$

Generally, this calculation of the neighborhood relevance subscore R_{Ni} will result in a higher value for listings that are located in neighborhoods in which bookings occur more frequently, and listings appear more often relative, to other neighborhoods in the same city. The

increased frequency of listings and bookings for a given neighborhood relative to others is taken indicative of a higher degree of intangible value of that neighborhood to consumers.

Distance Relevance

[0049] Generally, the distance relevance subscore D_i quantifies the distance between the location of the search query and the location of a listing, such that listings that are ranked further away are ranked lower using a non-linear function of distance. The reason a non-linear function is used is because greater distances impose more than a simple linear increase in inconvenience or cost to users; in other words, being ten miles away from a desired location is more than simply twice as inconvenience than being five miles away. In one embodiment, this distance *d* is determined from a center of a query city or neighborhood, as determined by geographical extent or population density. In another embodiment, an external data source may provide the geographical location from which the distance *d* is determined. In another embodiment, the distance *d* is measured based on a distance between the listing *i* and a user's current location, for example as provided by the client device 101 (e.g., a smart phone).

[0050] The distance relevance subscore D_i may be a sigmoidal, exponential, stepped, piecewise linear, or ~~other~~ any other type of function. In one embodiment, D_i is determined according to

$$D_i = \frac{a}{1 + e^{(d/b) - c}}$$

where ~~*d*~~ *a* is the distance between the location specified in the query, and *b*, *c*, and *d* are configurable constants.

Location Relevance Example

[0051] In one embodiment, the location relevance module 213 determines the location relevance score R_i for a listing *i* is determined according to:

$$R_i = \begin{cases} R_{Ci} \cdot (1 + R_{Ni}) & \text{if } Q_{C*} = C_i \text{ city has neighborhoods} \\ R_{Ci} \cdot (1 + D_i) & \text{if } Q_{C*} = C_i \text{ no neighborhoods} \\ R_{Ci} & \text{if city data available} \\ D_i^2 & \text{if no city data available} \end{cases}$$

where the city relevance subscore R_{Ci}, the neighborhood relevance subscore R_{Ni}, and the distance relevance subscore D_i are determined as described above. In alternate embodiments, other types of data not mentioned here may be used to determine location relevance score R_i in addition to or in place of the subscores described here.

[0052] Determining the location relevance score R_i according to this function allows the location relevance score R_i to vary depending upon the search query and the data available to the online booking system 111. Depending upon the search query, historical listing, booking,

Exemplary Method

[0057] FIG. 3 is a flow chart for ranking listings of goods or services using a location relevance score, according to one embodiment. In step 301, the online booking system 111 receives a search query from a client device, where the search query includes at least a query city, and in some cases a query state and query country as well. In step 303, the online booking system 111 accesses a number of stored listings for goods or services, as well as historical search queries and historical bookings that occurred subsequent to the search queries. In step 305, the online booking system 111 determines a location relevance score for each accessed listing. The location relevance score for a listing is based on at least one of a city relevance subscore, a neighborhood relevance subscore, and a distance subscore. In step 307, the online booking system 111 ranks the listings based at least in part on the location relevance score. The rankings may also be based on other factors, for example number of reviews provided by users for a listing, the number of bookings of the listing, the price of listing, and so on. In step 309, the online booking system 111 provides the listings to the client device in ranked order.

CLAIMS

1. A computer implemented method, comprising:

receiving a search query from a client device associated with a user, the search query including a query city;

accessing a plurality of listings available for booking by users, each of the listings located in one of a plurality of listing cities, at least one of the listing cities being different from the query city;

accessing a plurality of historical search queries including the query city, and a plurality of historical bookings in the plurality of listing cities;

determine, by a computer processor, a location relevance score for each of the listings based on a city relevance subscore, the city relevance subscore for a listing based on the query city, the listing city, the historical search queries, and the historical bookings;

ranking the listings based at least in part on the city relevance subscore; and

providing the listings in ranked order to the client device associated with the user.

2. The computer implemented method of claim 1, wherein the historical bookings include at least one booking located in one of the listing cities that is different from the query city.

3. The computer implemented method of claim 1,

wherein the city relevance subscore a listing is based on a first function if the query city is the same as the listing city, and

wherein the city relevance subscore is based on the first function and an additional weighting factor if the query city is different from the listing city.

4. The computer implemented method of claim 1, wherein the city relevance subscore for the listing is based on a number of historical bookings in the listing city that occurred during a same web browsing session as historical search queries including the query city.

5. The computer implemented method of claim 4, wherein the city relevance subscore for the listing is further based on a total number of historical bookings that occurred during a same web browsing session as historical search queries including the query city.

~~6. The computer implemented method of claim 1, wherein the city relevance subscore for the listing is based on a number of historical bookings in the listing city that occurred during a same web browing session as historical search queries including the query city.~~

7. The computer implemented method of claim ~~6~~1, wherein the city relevance subscore for the listing is further based on a total number of historical bookings in the listing city .

~~8~~7. The computer implemented method of claim 1, wherein the city relevance subscore for the listing is based on a number of listings in the listing city.

~~9~~8. The computer implemented method of claim 1, wherein the city relevance subscore for the listing is based on a number of listings in the listing city and a total number of listings in the plurality of listings.

~~10~~9. A computer implemented method, comprising:

receiving a search query from a client device associated with user, the search query including a query city;

accessing a plurality of listings available for booking by users, each of the listings located in one of a plurality of listing neighborhoods in the query city;

accessing a plurality of historical bookings in the query city;

determine, by a computer processor, a location relevance score for each of the listings based on a neighborhood relevance subscore, the neighborhood relevance subscore for one of the listings based on the query city, the historical bookings, and the other accessed listings;

ranking the listings based at least in part on the neighborhood relevance subscore; and

providing the listings in ranked order to the client device associated with the user.

~~11~~10. The computer implemented method of claim ~~10~~9, wherein the historical bookings are for listings from a plurality of listing neighborhoods in the query city.

~~12~~11. The computer implemented method of claim ~~10~~9, wherein the neighborhood relevance subscore for the listing is based on a number of historical bookings in the listing neighborhood.

~~13~~12. The computer implemented method of claim ~~12~~11, wherein the neighborhood relevance subscore for the listing is further based on a total number of historical listings in ~~the~~ a listing city of the listing neighborhood.

~~14~~13. The computer implemented method of claim ~~10~~9, wherein the neighborhood relevance subscore for a listing is based on a number of available listings in the listing neighborhood.

~~15~~14. The computer implemented method of claim ~~14~~13, wherein the neighborhood relevance subscore for a listing is further based on a total number of available listings in ~~the~~ a listing city of the listing neighborhood.

~~16~~15. The computer implemented method of claim ~~10~~9, comprising:

accessing a plurality of historical listings that are no longer available for booking by users, each of the historical listings located in one of the listing neighborhoods in the query city; and