US 20060224554A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2006/0224554 A1

Bailey et al. (43) Pub. Date: Oct. 5, 2006

(54) QUERY REVISION USING KNOWN HIGHLY-RANKED QUERIES

(76) Inventors: **David R. Bailey**, Palo Alto, CA (US); **Alexis J. Battle**, Palo Alto, CA (US); **David Ariel Cohn**, Palo Alto, CA (US); **Barbara Engelhardt**, San Francisco, CA (US); **P. Pandurang Nayak**, Palo Alto, CA (US)

Correspondence Address:
GOOGLE / FENWICK
SILICON VALLEY CENTER
801 CALIFORNIA ST.
MOUNTAIN VIEW, CA 94041 (US)

(21) Appl. No.: **11/285,814**

(22) Filed: **Nov. 22, 2005**

Related U.S. Application Data

(63) Continuation-in-part of application No. 11/094,814, filed on Mar. 29, 2005.
Continuation-in-part of application No. 11/096,198, filed on Mar. 30, 2005.
Continuation-in-part of application No. 11/095,920, filed on Mar. 30, 2005.

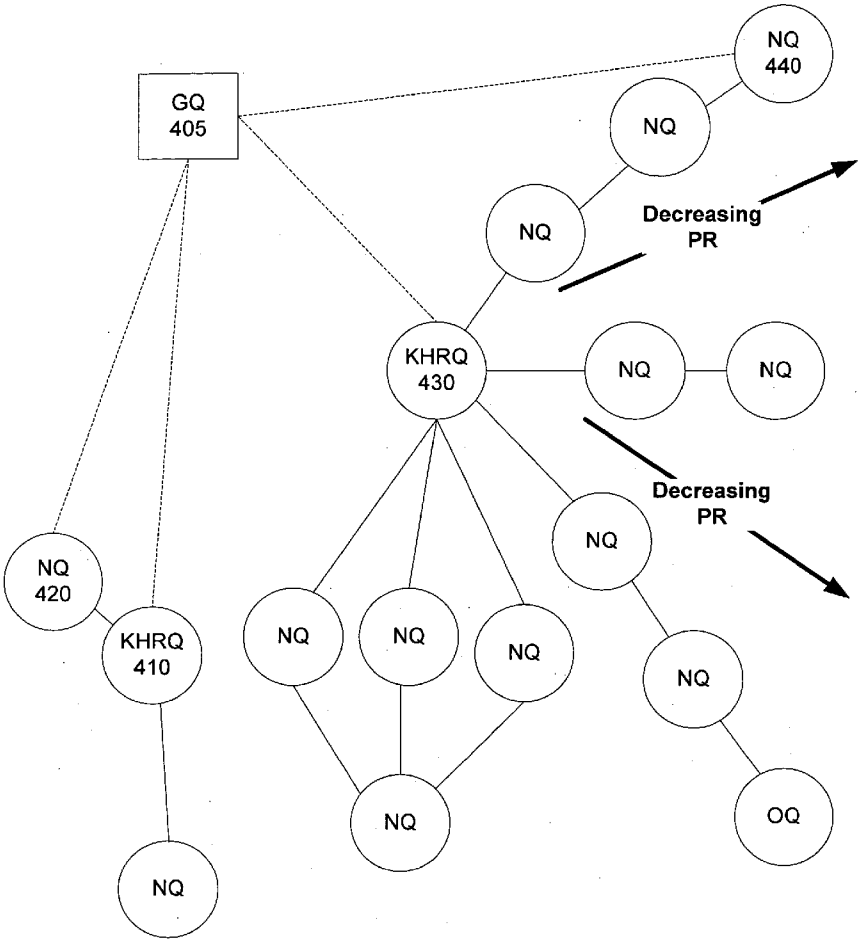
Publication Classification

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/1**

(57) **ABSTRACT**

An information retrieval system includes a query revision architecture providing one or more query revisers, each of which implements a query revision strategy. A query rank reviser suggests known highly-ranked queries as revisions to a first query by initially assigning a rank to all queries, and identifying a set of known highly-ranked queries (KHRQ). Queries with a strong probability of being revised to a KHRQ are identified as nearby queries (NQ). Alternative queries that are KHRQs are provided as candidate revisions for a given query. For alternative queries that are NQs, the corresponding known highly-ranked queries are provided as candidate revisions.



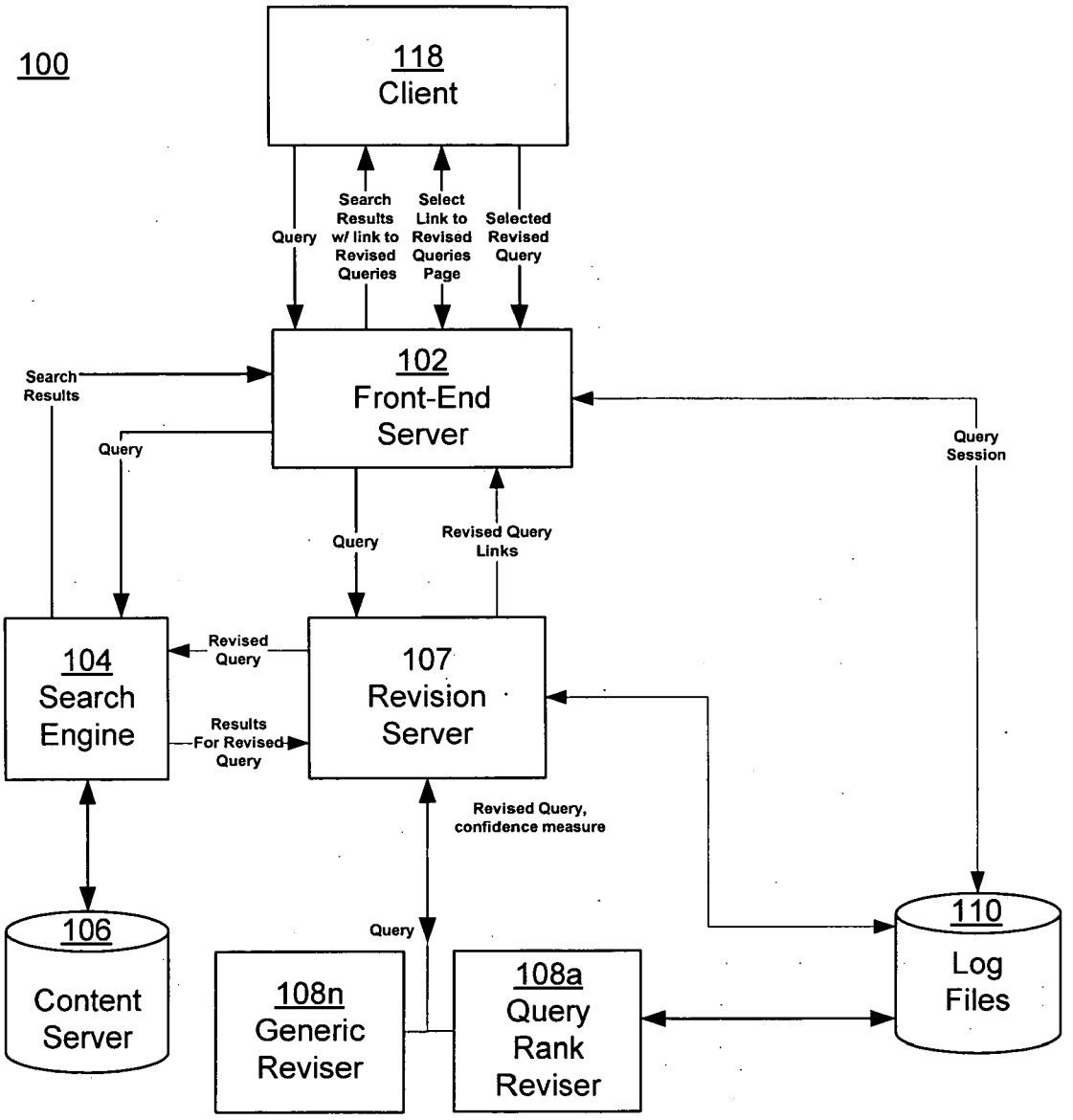

100
118
Client
Query
Search Results w/ link to Revised Queries
Select Link to Revised Queries Page
Selected Revised Query
102
Front-End Server
Search Results
Query
Query Session
Query
Revised Query Links
104
Search Engine
Revised Query
Results For Revised Query
107
Revision Server
Revised Query, confidence measure
106
Content Server
Query
108n
Generic Reviser
108a
Query Rank Reviser
110
Log Files

FIG. 1a

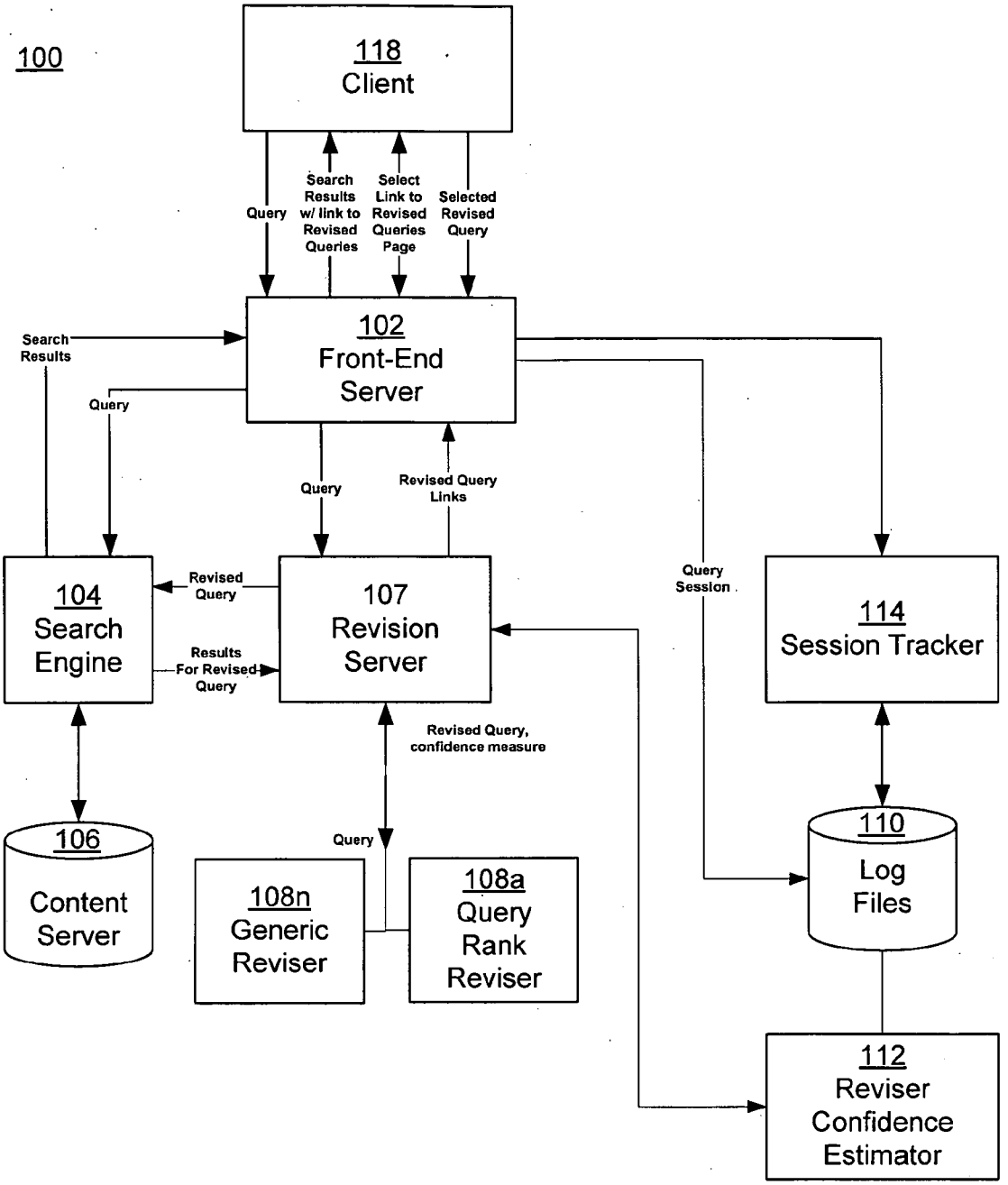

100
118
Client
Query
Search Results w/ link to Revised Queries
Select Link to Revised Queries Page
Selected Revised Query
102
Front-End Server
Search Results
Query
Query
Revised Query Links
104
Search Engine
Revised Query
Results For Revised Query
107
Revision Server
Query Session
114
Session Tracker
Revised Query, confidence measure
106
Content Server
Query
108n
Generic Reviser
108a
Query Rank Reviser
110
Log Files
112
Reviser Confidence Estimator

FIG. 1b

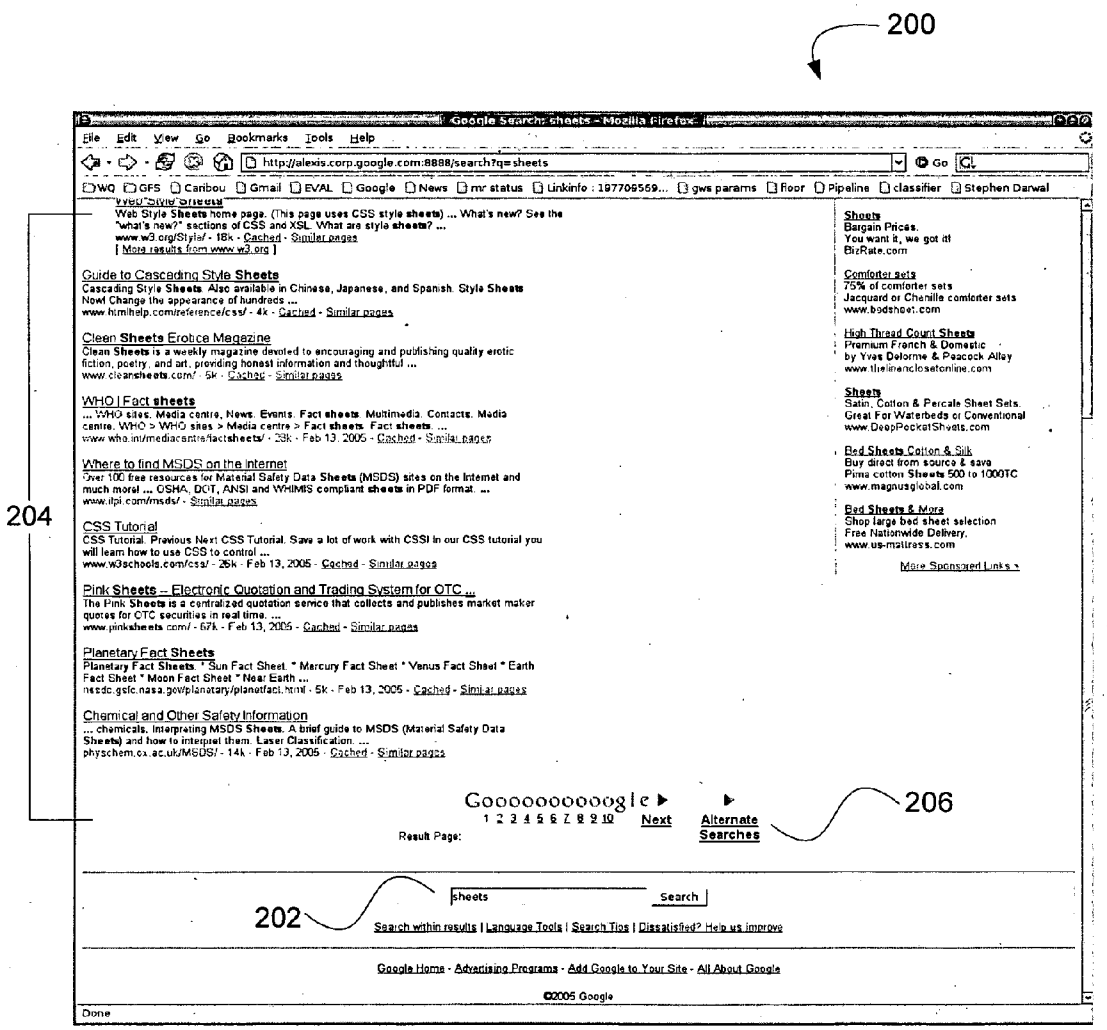

200
Google Search: sheets - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://alexis.corp.google.com:8888/search?q=sheets
Web Style Sheets
Web Style Sheets home page. (This page uses CSS style sheets) ... What's new? See the "what's new?" sections of CSS and XSL. What are style sheets? ...
www.w3.org/Style/ - 18k - Cached - Similar pages
[More results from www.w3.org]
Guide to Cascading Style Sheets
Cascading Style Sheets. Also available in Chinese, Japanese, and Spanish. Style Sheets Now! Change the appearance of hundreds ...
www.htmlhelp.com/reference/css/ - 4k - Cached - Similar pages
Clean Sheets Erotica Magazine
Clean Sheets is a weekly magazine devoted to encouraging and publishing quality erotic fiction, poetry, and art, providing honest information and thoughtful ...
www.cleansheets.com/ - 5k - Cached - Similar pages
WHO | Fact sheets
... WHO sites. Media centre. News. Events. Fact sheets. Multimedia. Contacts. Media centre. WHO > WHO sites > Media centre > Fact sheets. Fact sheets. ...
www.who.int/mediacentre/factsheets/ - 33k - Feb 13, 2005 - Cached - Similar pages
Where to find MSDS on the internet
Over 100 free resources for Material Safety Data Sheets (MSDS) sites on the Internet and much more! ... OSHA, DOT, ANSI and WHMIS compliant sheets in PDF format. ...
www.itpi.com/msds/ - Similar pages
CSS Tutorial
CSS Tutorial. Previous Next CSS Tutorial. Save a lot of work with CSS! In our CSS tutorial you will learn how to use CSS to control ...
www.w3schools.com/css/ - 26k - Feb 13, 2005 - Cached - Similar pages
Pink Sheets – Electronic Quotation and Trading System for OTC ...
The Pink Sheets is a centralized quotation service that collects and publishes market maker quotes for OTC securities in real time. ...
www.pinksheets.com/ - 67k - Feb 13, 2005 - Cached - Similar pages
Planetary Fact Sheets
Planetary Fact Sheets. * Sun Fact Sheet. * Mercury Fact Sheet * Venus Fact Sheet * Earth Fact Sheet * Moon Fact Sheet * Near Earth ...
nssdc.gsfc.nasa.gov/planetary/planetfact.html - 5k - Feb 13, 2005 - Cached - Similar pages
Chemical and Other Safety Information
... chemicals. Interpreting MSDS Sheets. A brief guide to MSDS (Material Safety Data Sheets) and how to interpret them. Laser Classification. ...
physchem.ox.ac.uk/MSDS/ - 14k - Feb 13, 2005 - Cached - Similar pages
Sheets
Bargain Prices.
You want it, we got it!
BizRate.com
Comforter sets
75% of comforter sets
Jacquard or Chenille comforter sets
www.bedsheet.com
High Thread Count Sheets
Premium French & Domestic
by Yves Delorme & Peacock Alley
www.thelinencloset.com
Sheets
Satin, Cotton & Percale Sheet Sets.
Great For Waterbeds or Conventional
www.DeepPocketSheets.com
Bed Sheets Cotton & Silk
Buy direct from source & save
Pima cotton Sheets 500 to 1000TC
www.magnusglobal.com
Bed Sheets & More
Shop large bed sheet selection
Free Nationwide Delivery.
www.us-mattress.com
More Sponsored Links »
204
Goooooooooogle
1 2 3 4 5 6 7 8 9 10 Next
Result Page:
Alternate Searches
206
sheets
Search
202
Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve
Google Home - Advertising Programs - Add Google to Your Site - All About Google
©2005 Google
Done

FIG. 2

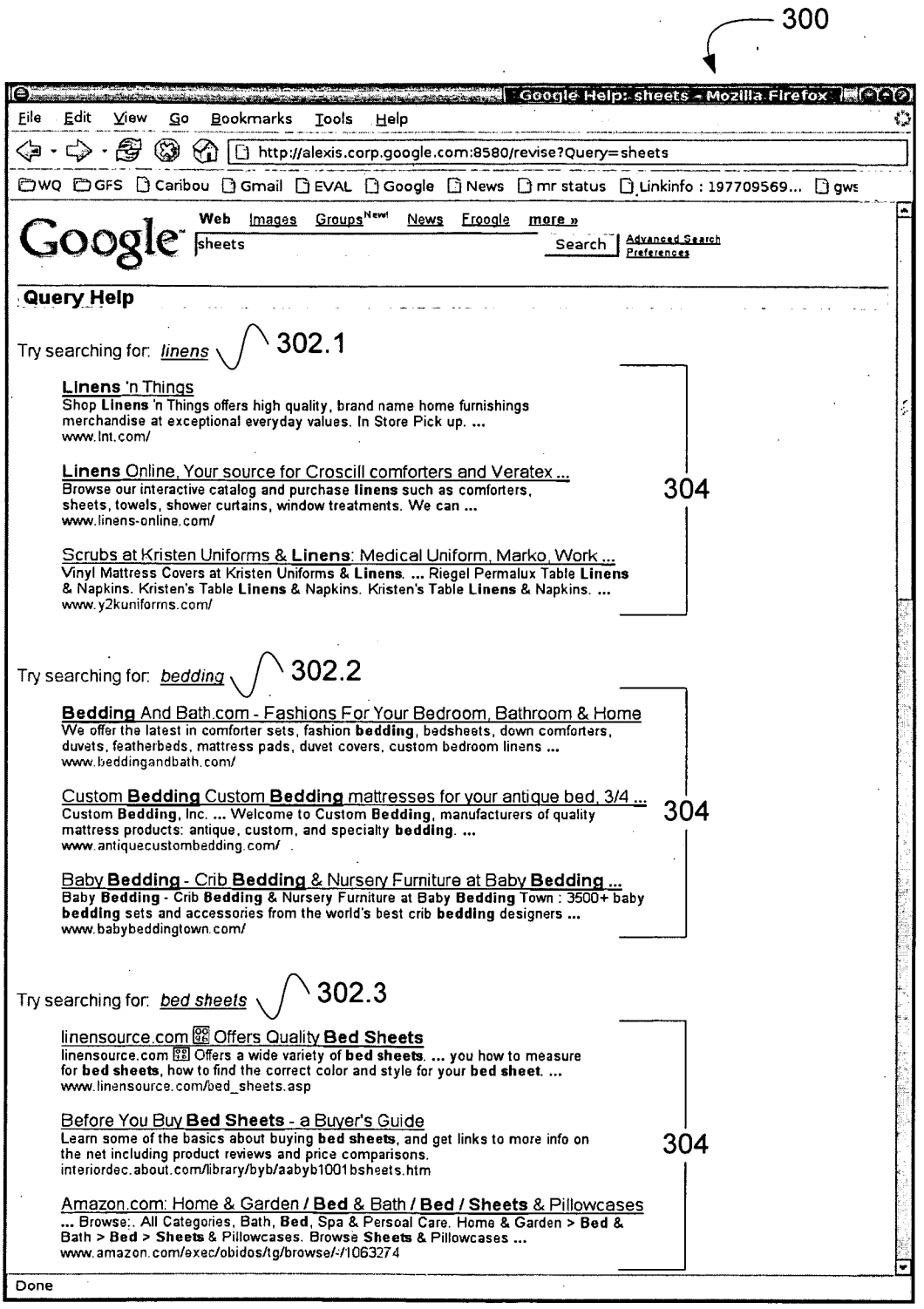

300
Google Help: sheets - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://alexis.corp.google.com:8580/revise?Query=sheets
WQ GFS Caribou Gmail EVAL Google News mr status Linkinfo : 197709569... gws
Google
Web Images Groups New! News Froogle more »
sheets
Search
Advanced Search Preferences
Query Help
Try searching for: linens
302.1
Linens 'n Things
Shop Linens 'n Things offers high quality, brand name home furnishings merchandise at exceptional everyday values. In Store Pick up. ...
www.lnt.com/
Linens Online, Your source for Croscill comforters and Veratex ...
Browse our interactive catalog and purchase linens such as comforters, sheets, towels, shower curtains, window treatments. We can ...
www.linens-online.com/
Scrubs at Kristen Uniforms & Linens: Medical Uniform, Marko, Work ...
Vinyl Mattress Covers at Kristen Uniforms & Linens. ... Riegel Permalux Table Linens & Napkins. Kristen's Table Linens & Napkins. Kristen's Table Linens & Napkins. ...
www.y2kuniforms.com/
304
Try searching for: bedding
302.2
Bedding And Bath.com - Fashions For Your Bedroom, Bathroom & Home
We offer the latest in comforter sets, fashion bedding, bedsheets, down comforters, duvets, featherbeds, mattress pads, duvet covers, custom bedroom linens ...
www.beddingandbath.com/
Custom Bedding Custom Bedding mattresses for your antique bed, 3/4 ...
Custom Bedding, Inc. ... Welcome to Custom Bedding, manufacturers of quality mattress products: antique, custom, and specialty bedding. ...
www.antiquecustombedding.com/
Baby Bedding - Crib Bedding & Nursery Furniture at Baby Bedding ...
Baby Bedding - Crib Bedding & Nursery Furniture at Baby Bedding Town : 3500+ baby bedding sets and accessories from the world's best crib bedding designers ...
www.babybeddingtown.com/
304
Try searching for: bed sheets
302.3
linensource.com Offers Quality Bed Sheets
linensource.com Offers a wide variety of bed sheets. ... you how to measure for bed sheets, how to find the correct color and style for your bed sheet. ...
www.linensource.com/bed_sheets.asp
Before You Buy Bed Sheets - a Buyer's Guide
Learn some of the basics about buying bed sheets, and get links to more info on the net including product reviews and price comparisons.
interiordec.about.com/library/byb/aabyb1001bsheets.htm
Amazon.com: Home & Garden / Bed & Bath / Bed / Sheets & Pillowcases
... Browse:. All Categories, Bath, Bed, Spa & Persoal Care. Home & Garden > Bed & Bath > Bed > Sheets & Pillowcases. Browse Sheets & Pillowcases ...
www.amazon.com/exec/obidos/tg/browse/-/1063274
304
Done

FIG. 3

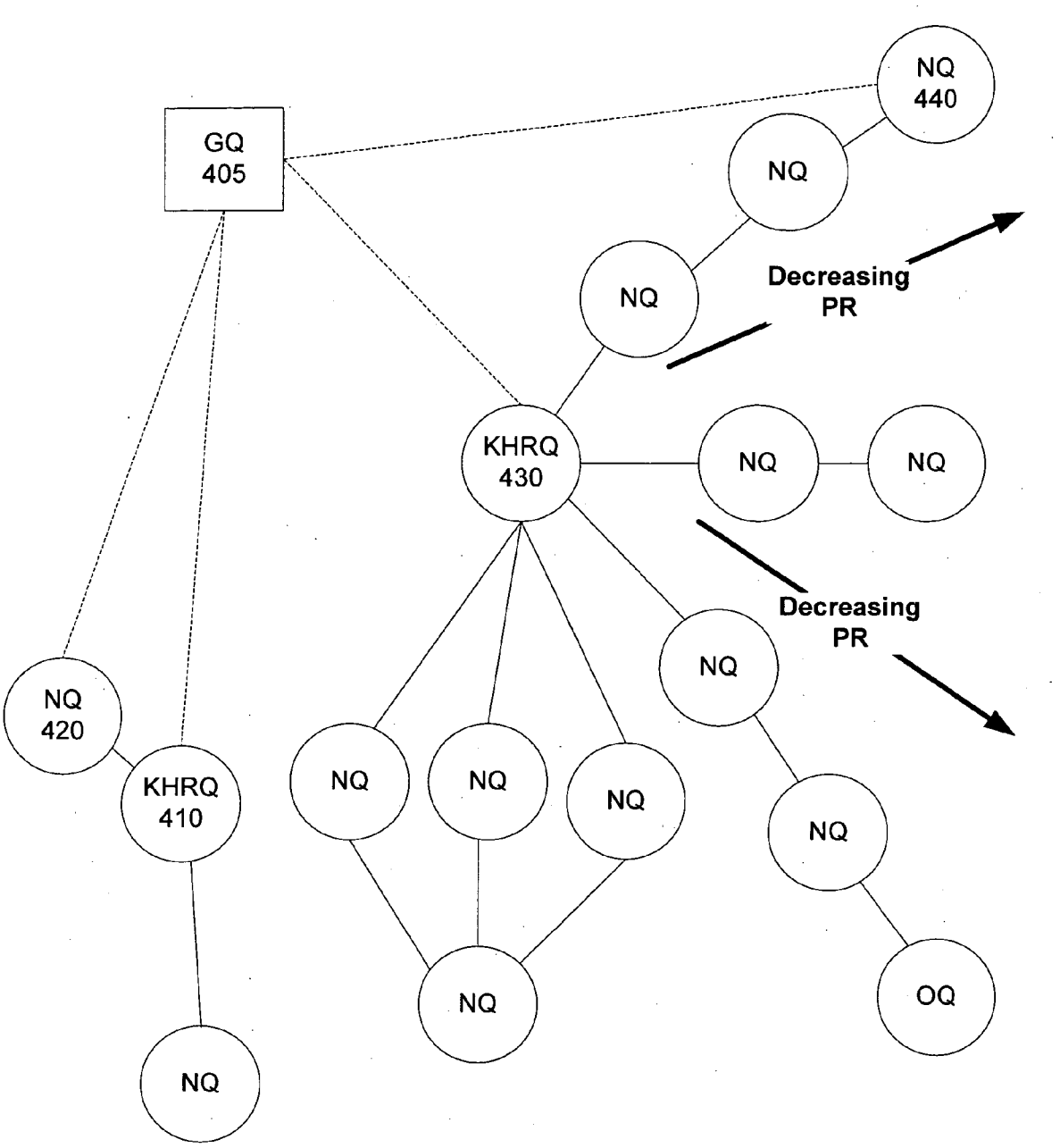

GQ
405
NQ
440
NQ
NQ
Decreasing
PR
KHRQ
430
NQ
NQ
Decreasing
PR
NQ
NQ
420
KHRQ
410
NQ
NQ
NQ
NQ
NQ
OQ
NQ

FIG. 4

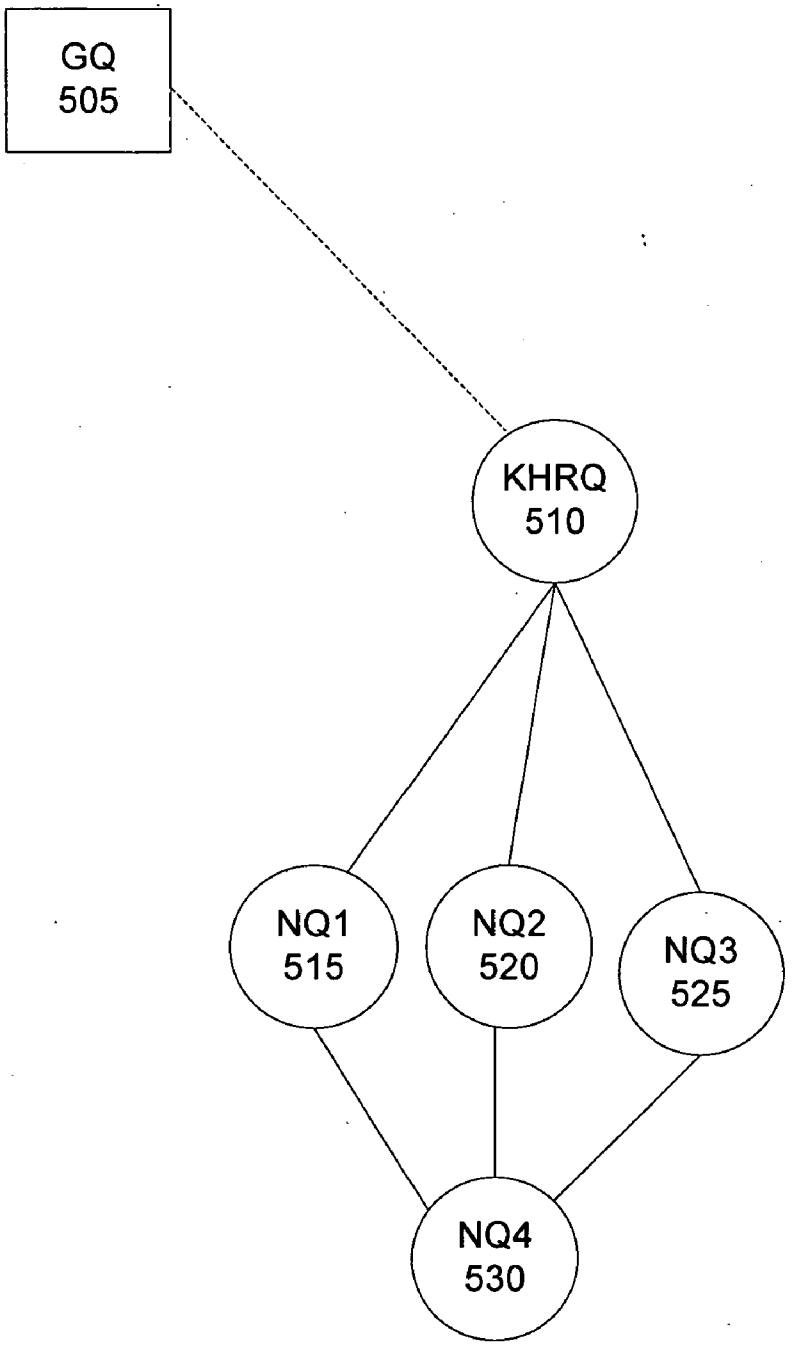


FIG. 5

QUERY REVISION USING KNOWN HIGHLY-RANKED QUERIES

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims priority under 35 U.S.C. § 120 from U.S. application Ser. No. 11/094,814, filed on Mar. 29, 2005, entitled "Integration Of Multiple Query Revision Models," U.S. application Ser. No. 11/096,198, filed on Mar. 30, 2005, entitled "Estimating Confidence For Query Revision Models," U.S. application Ser. No. 11/095,920, filed on Mar. 30, 2005, entitled "Empirical Validation Of Suggested Alternative Queries," and is related to:

[0002] U.S. application Ser. No. 10/676,571, filed on Sep. 30, 2003, entitled "Method and Apparatus for Characterizing Documents Based on Clusters of Related Words;"

[0003] U.S. application Ser. No. 10/734,584, filed Dec. 15, 2003, entitled "Large Scale Machine Learning Systems and Methods;"

[0004] U.S. application Ser. No. 10/749,440, filed on Dec. 31, 2003, entitled "Methods and Systems for Assisted Network Browsing;" and

[0005] U.S. application Ser. No. 10/878,926, "Systems and Methods for Deriving and Using an Interaction Profile," filed on Jun. 28, 2004,"

[0006] each of which is incorporated herein by reference.

FIELD

[0007] The present invention relates to information retrieval systems generally, and more particularly to systems and methods for revising user queries.

BACKGROUND

[0008] Information retrieval systems, as exemplified by Internet search engines, are generally capable of quickly providing documents that are generally relevant to a user's query. Search engines may use a variety of statistical measures of term and document frequency, along with linkages between documents and between terms to determine the relevance of document to a query. A key technical assumption underlying most search engine designs is that a user query accurately represents the user's desired information goal.

[0009] In fact, users typically have difficulty formulating good queries. Often, a single query does not provide desired results, and users frequently enter a number of different queries about the same topic. These multiple queries will typically include variations in the breadth or specificity of the query terms, guessed names of entities, variations in the order of the words, the number of words, and so forth, sometimes forming long chains of queries before reaching the desired result set. Because different users have widely varying abilities to successfully revise their queries, various automated methods of query revision have been proposed.

[0010] Most commonly, query refinement is used to automatically generate more precise (i.e., narrower) queries from a more general query. Query refinement is primarily useful when users enter over-broad queries whose top results include a superset of documents related to the user's information needs. For example, a user wanting information on the Mitsubishi Galant automobile might enter the query "Mitsubishi," which is overly broad, as the results will cover the many different Mitsubishi companies, not merely the automobile company. Thus, refining the query would be desirable (though difficult here because of the lack of additional context to determine the specific information need of the user).

[0011] However, query refinement is not useful when users enter overly specific queries, where the right revision is to broaden the query, or when the top results are unrelated to the user's information needs. For example, the query "Mitsubishi Galant information" might lead to poor results (in this case, too few results about the Mistubishi Galant automobile) because of the term "information." In this case, the right revision is to broaden the query to "Mitsubishi Galant." Thus, while query refinement works in some situations, there are a large number of situations where a user's information needs are best met by using other query revision techniques.

[0012] Another query revision strategy uses synonym lists or thesauruses to expand the query to capture a user's potential information need. As with query refinement, however, query expansion is not always the appropriate way to revise the query, and the quality of the results is very dependent on the context of the query terms.

SUMMARY

[0013] An information retrieval system includes a query revision architecture that provides one or more different query revisers, each of which implements its own query revision strategy. Each query reviser evaluates a user query to determine one or more potential revised queries of the user query. A revision server interacts with the query revisers to obtain the potential revised queries. The revision server also interacts with a search engine in the information retrieval system to obtain for each potential revised query a set of search results. The revision server selects one or more of the revised queries for presentation to the user, along with a subset of search results for each of the selected revised queries. The user is thus able to observe the quality of the search results for the revised queries, and then select one of the revised queries to obtain a full set of search results for the revised query according to one embodiment.

[0014] A system and method use session-based user data to more correctly capture a user's potential information need based on analysis of strings of queries other users have formed in the past. To accomplish this, revised queries are provided based on data collected from many individual user sessions. For example, such data may include click data, explicit user data, or hover data. For a description of user feedback using hover data, see U.S. application Ser. No. 10/749,440, filed on Dec. 31, 2003, entitled "Methods and Systems for Assisted Network Browsing," which is incorporated herein by reference.

[0015] In one embodiment, a query rank reviser suggests one or more known highly-ranked queries as a revision to a first query. Initially, a query rank is assigned to all queries. The query rank reviser creates a table of queries and respective query ranks, identifying the highest ranked que-

ries as known highly-ranked queries (KHRQ). Queries with a strong probability of being revised to a KHRQ are identified as nearby queries (NQ), a pointer from each NQ to the corresponding KHRQ(s) is stored, and the KHRQs and NQs queries are indexed.

[0016] For a given query, the query rank reviser determines a revision probability with respect to the indexed queries. Next, a revision score (RS) is calculated for each indexed query using the revision probability and query rank for the indexed query. Then the indexed queries with the highest revision scores are retrieved as alternative queries. Alternative queries that are KHRQs are provided as candidate revisions and for alternative queries that are NQs, the corresponding known highly-ranked query are provided as candidate revisions, using the pointers stored in the index.

[0017] The present invention is next described with respect to various figures, diagrams, and technical information. The figures depict various embodiments of the present invention for purposes of illustration only. One skilled in the art will readily recognize from the following discussion that alternative embodiments of the illustrated and described structures, methods, and functions may be employed without departing from the principles of the invention.

BRIEF DESCRIPTION

[0018] **FIG. 1** is a system diagram of an embodiment of an information retrieval system providing for query revision according to one embodiment of the present invention.

[0019] **FIG. 2** is an illustration of a sample results page to an original user query according to one embodiment of the present invention.

[0020] **FIG. 3** is an illustration of a sample revised queries page according to one embodiment of the present invention.

[0021] **FIG. 4** illustrates a graphed topology of queries according to one embodiment of the present invention.

[0022] **FIG. 5** illustrates a graphed topology of queries according to another embodiment of the present invention.

DETAILED DESCRIPTION

System Overview

[0023] **FIG. 1***a* illustrates a system **100** in accordance with one embodiment of the present invention. System **100** comprises a front-end server **102**, a search engine **104** and associated content server **106**, a revision server **107**, and one or more query revisers **108**. During operation, a user accesses the system **100** via a conventional client **118** over a network (such as the Internet, not shown) operating on any type of client computing device, for example, executing a browser application or other application adapted to communicate over Internet related protocols (e.g., TCP/IP and HTTP). While only a single client **118** is shown, the system **100** can support a large number of concurrent sessions with many clients. In one implementation, the system **100** operates on high performance server class computers, and the client device **118** can be any type of computing device. The details of the hardware aspects of server and client computers is well known to those of skill in the art and is not further described here.

[0024] The front-end server **102** is responsible for receiving a search query submitted by the client **118**. The front-end server **102** provides the query to the search engine **104**, which evaluates the query to retrieve a set of search results in accordance with the search query, and returns the results to the front-end server **102**. The search engine **104** communicates with one or more of the content servers **106** to select a plurality of documents that are relevant to user's search query. A content server **106** stores a large number of documents indexed (and/or retrieved) from different websites. Alternately, or in addition, the content server **106** stores an index of documents stored on various websites. "Documents" are understood here to be any form of indexable content, including textual documents in any text or graphics format, images, video, audio, multimedia, presentations, web pages (which can include embedded hyperlinks and other metadata, and/or programs, e.g., in Javascript), and so forth. In one embodiment, each indexed document is assigned a page rank according to the document's link structure. The page rank serves as a query-independent measure of the document's importance. An exemplary form of page rank is described in U.S. Pat. No. 6,285,999, which is incorporated herein by reference. The search engine **104** assigns a score to each document based on the document's page rank (and/or other query-independent measures of the document's importance), as well as one or more query-dependent signals of the document's importance (e.g., the location and frequency of the search terms in the document).

[0025] The front-end server **102** also provides the query to the revision server **107**. The revision server **107** interfaces with one or more query revisers **108**, each of which implements a different query revision strategy or set of strategies. In one embodiment, the query revisers **108** include a query rank reviser **108***a*. The revision server **107** provides the query to each reviser **108**, and obtains in response from each reviser **108** one or more potential revised queries (called 'potential' here, since they have not been adopted at this point by the revision server **107**). The system architecture is specifically designed to allow any number of different query revisers **108** to be used, for poor performing query revisers **108** to be removed, and for new query revisers **108** (indicated by generic reviser **108***n*) to be added as desired in the future. This gives the system **100** particular flexibility, and also enables it to be customized and adapted for specific subject matter domains (e.g., revisers for use in domains like medicine, law, etc.), enterprises (revisers specific to particular business fields or corporate domains, for internal information retrieval systems), or for different languages (e.g., revisers for specific languages and dialects).

[0026] Preferably, each revised query is associated with a confidence measure representing the probability that the revision is a good revision, i.e., that the revised query will produce results more relevant to the user's information needs than the original query. Thus, each potential revised query can be represented by the tuple (Ri, Ci), where R is a potential revised query, and C is the confidence measure associated with the revised query. In one embodiment, these confidence measures are manually estimated beforehand for each revision strategy of each reviser **108**. The measures can be derived from analysis of the results of sample queries and revised queries under test. In other embodiments, one or more of the revisers **108** may dynamically generate a confidence measure (e.g., at run time) for one or more of its potential revised queries. The assignment of confidence measures may be performed by other components (e.g., the

revision server **107**), and may take into account both query-dependent and query-independent data.

[**0027**] The revision server **107** can select one or more (or all) of the potential revised queries, and provide these to the search engine **104**. The search engine **104** processes a revised query in the same manner as normal queries, and provides the results of each submitted revised query to the revision server **107**. The revision server **107** evaluates the results of each revised query, including comparing the results for the revised query with the results for the original query. The revision server **107** can then select one or more of the revised queries as being the best revised queries (or at least revised queries that are well-suited for the original query), as described below.

[**0028**] The revision server **107** receives all of the potential revised queries R, and sorts them by their associated confidence measures C, from highest to lowest confidence. The revision server **107** iterates through the sorted list of potential revised queries, and passes each potential revised query to the search engine **104** to obtain a set of search results. (Alternatively, the revision server **107** may first select a subset of the potential revised queries, e.g., those with a confidence measure above a threshold level). In some cases the top search results may already have been fetched (e.g., by a reviser **108** or the revision server **107**) while executing a revision strategy or in estimating confidence measures, in which case the revision server **107** can use the search results so obtained.

[**0029**] For each potential revised query, the revision server **107** decides whether to select the potential revised query or discard it. The selection can depend on an evaluation of the top N search results for the revised query, both independently and with respect to the search results of the original query. Generally, a revised query should produce search results that are more likely to accurately reflect the user's information needs than the original query. Typically the top ten results are evaluated, though more or less results can be processed, as desired.

[**0030**] In one embodiment, a potential revised query is selected if the following conditions hold:

[**0031**] i) The revised query produces at least a minimum number of search results. For example, setting this parameter to 1 will discard all (and only) revisions with no search results. The general range of an acceptable minimum number of results is 1 to 100.

[**0032**] ii) The revised query produces a minimum number of "new" results in a revision's top results. A result is "new" when it does not also occur in the top results of the original query or a previously selected revised query. For example, setting this parameter to 2 would require each selected revision to have at least two top results that do not occur in the top results of any previously selected revised query or in the top results of the original query. This constraint ensures that there is a diversity of results in the selected revisions, maximizing the chance that at least one of the revisions will prove to be useful. For example, as can be seen in **FIG. 3**, the top three results **304** for each revised query are distinct from the other result sets. This gives the user a broad survey of search results that are highly relevant to the revised queries.

[**0033**] iii) A maximum number of revised queries have not yet been selected. In other words, when a maximum number of revised queries have already been selected, then all remaining revised queries are discarded. In one embodiment, the maximum number of revised queries is set at 4. In another embodiment, the maximum number of revised queries is set between 2 and 10.

[**0034**] The results of the foregoing selection parameters are a set of selected revised queries that will be included on the revised queries page **300**. The revision server **107** constructs a link to this page, and provides this link to the front-end server **102**, as previously discussed. The revision server **107** determines the order and layout of the revised queries on the revised queries page **300**. The revised queries are preferably listed in order of their confidence measures (from highest to lowest).

[**0035**] The front-end server **102** includes the provided links in a search results page, which is then transmitted to the client **118**. The user can then review the search results to the original query, or select the link to the revised queries page, and thereby view the selected revised queries and their associated results.

Presentation of Revised Queries

[**0036**] **FIG. 2** illustrates a sample results page **200** provided to a client **118**. In this simple implementation, the search results **200** page includes the original query **202** of [sheets] along with the results **204** to this query. A link **206** to a set of revised queries is included at the bottom of the page **200**. The user can then click on the link **206**, and access the page of revised queries. An example page **300** is shown in **FIG. 3**. Here, the top three revised queries are presented, as shown by revised query links **302.1**, **302.2**, and **302.3** for the revised queries of [linens], [bedding], and [bed sheets], respectively. Below each revised query link **302** are the top three search results **304** for that query.

[**0037**] There are various benefits to providing the revised queries on a separate page **300** from the original results page **200**. First, screen area is a limited resource, and thus listing the revised queries by themselves (without a preview of their associated results), while possible, is less desirable because the user does not see revised queries in the context of their results. By placing the revised queries on a separate page **300**, the user can see the best revised queries and their associated top results, enabling the user to choose which revised query appears to best meet their information needs, before selecting the revised query itself. While it would be possible to include both the results of the original query and the revised queries on a single (albeit long) page, this approach would either require to the user to scroll down the page to review all of the revised queries, or would clutter the initially visible portion of the page. Instead, in the preferred embodiment illustrated in **FIGS. 2 and 3**, the user can see results associated with query revisions, click on each revised query link **302**, and access the entire set of search results for the selected revised query. In many cases this approach will also be preferable to automatically using the revised queries to obtain search results and automatically presenting them to the user (e.g., without user selection or interaction). In the example query revision described in conjunction with the query rank reviser, the benefits of this method are clear: [Britney Spears] would be a suggested query because of its high query rank, but does not get the user the desired information. Thus, it is helpful to display the results and query to the user for selection. In addition, this approach has

the added benefit of indirectly teaching the user how to create better queries, by showing the best potential revisions. In another embodiment, the revision server **107** can force the query revisions to be shown on the original result page **200**, for example, in a separate window or within the original result page **200**.

[**0038**] The method of displaying additional information (e.g., search results **304**), about query revisions to help users better understand the revisions can also be used on the main results page **200**. This is particularly useful when there is a single very high quality revised query (or a small number of very high quality revisions) such as is the case with revisions that correct spellings. Spell corrected revised queries can be shown on the results page **200**, along with additional information such as title, URL, and snippet of the top results to help the user in determining whether or not the spell correction suggestion is a good one.

[**0039**] In another embodiment, revision server **107** uses the confidence measures to determine whether to show query revisions at all, and if so, how prominently to place the revisions or the link thereto. This embodiment is discussed below.

Query Revising

[**0040**] Referring again to **FIG. 1***a*, one embodiment of the query rank reviser **108***a* is now described. The rank reviser **108***a* can use any suitable method to suggest known highly-ranked queries that might better capture the user's information need based on analysis of chains of revisions to queries made by other users in the past. In general, a highly ranked query is one that occurs frequently relative to other queries, but that is revised infrequently relative to its occurrences. That users only infrequently revise such queries indicates that the results provided by such queries adequately match the users' information needs.

[**0041**] In one embodiment, the highly-ranked queries are identified as follows. Initially, the query rank reviser **108***a* assigns a query rank to all queries stored in log files **110**. Query rank as used herein is defined using the occurrence frequency of a query (QF) and user satisfaction (US) with the query, e.g., the product of QF and US, i.e., (QR=QF× US). In one embodiment, user satisfaction is measured as inverse revision frequency (IRF). In other words, user satisfaction increases as revision frequency decreases. Revision frequency in one embodiment is defined as the number of times a query is revised divided by the total number of occurrences of the query. Thus, in this embodiment, query rank is defined as query occurrence frequency-inverse revision frequency (QF-IRF); again, reflecting that highly ranked queries occur frequently, but are revised infrequently.

[**0042**] In another embodiment, user satisfaction is defined by the quality of the query. In one embodiment, a quality score for a query is estimated from user click behavior data estimating the length of clicks on search results. One such method for determining quality scores is the use of interaction profiles, as described in U.S. application Ser. No. 10/878,926, "Systems and Methods for Deriving and Using an Interaction Profile," filed on Jun. 28, 2004, which is incorporated by reference. The quality score calculation is stored, for example, in log files **110**. Quality scores are based on the estimated duration of a first selection of a search result, e.g., a first click on a search result. The duration of a given click is estimated from the times at which a first and subsequent selections occurred on search results, which times may be stored with other user session query data, for example in the log files **110**. Scoring includes assigning search results in which the user did not select/click on a search result a score of zero, and proceeds along an S-curve applied to the duration between the first click and a subsequent click, with longer clicks approaching a quality score of 1. In one embodiment, the formula for the curve is $1/(1+e^{-(x-40s)/10s})$, wherein x is the duration between clicks, and the inflection points are 20 seconds corresponding to a quality score of 0.1, 40 seconds corresponding to a quality score of 0.5, and 60 seconds corresponding to a quality score 0.9. In other embodiments, different curves are used in accordance with click durations believed to represent user satisfaction. For example, stretching the curve to the right would do a better job of rewarding clicks with very long durations, at the cost of reduced discrimination between short-duration clicks and no click at all. Clicks on unrelated content, for example banner ads, are excluded from the click analysis. In another embodiment, all result clicks for a query, rather than just the first, are collected. Thus, in this embodiment, query rank is defined as query occurrence frequency (QF) times quality (Q).

[**0043**] Query occurrence frequency (QF) is defined as the frequency per unit time, for example frequency per hour according to one embodiment. Thus, as query occurrence frequency and inverse document frequency or quality increase, query rank increases toward **1**. In other embodiments, query occurrence frequency may be defined in different ways.

[**0044**] Next, the query rank reviser **108***a* creates a table of queries and respective query ranks. From this data, the query rank reviser **108***a* identifies a subset of all queries as known highly-ranked queries (KHRQ). A known highly-ranked query is a query known to have a high query rank as described above, and as listed in the table of queries. In another embodiment, known highly-ranked queries are defined as the top X queries, e.g., the top 5,000 queries.

[**0045**] The query rank reviser **108***a* then identifies nearby queries (NQ), which are queries with a strong probability of revision (PR) to a KHRQ, as measured by the similarity between the NQ and a KHRQ. Similarity can be determined based on semantic similarity, syntactic similarity, behavioral similarity, or any combination thereof. In one embodiment, the similarity is behavioral similarity. In another embodiment, similarity includes the semantic similarity between the queries, for example considering factors such as lexical similarity and/or overlap in word clusters for the respective queries. In yet another embodiment, similarity includes the syntactic similarity between the queries, for example considering factors such as edit distance, term overlap, or other technique(s) commonly used in information retrieval. In yet another embodiment, similarity includes both semantic and syntactic similarity. In one embodiment, scoring similarity includes assigning a similarity score between 0 and 1, with more similar queries approaching a score of 1. For example, queries for which a word is misspelled by one character (e.g., [Brittney Spears] versus [Britney Spears]) have a high similarity score (e.g., 0.95), whereas a query that has a small term overlap (e.g., [When Harry Met Sally] versus [Metropolitan Life]) have a low similarity score (e.g., 0.15). In embodiments in which greater than one type of similarity is

used, a parameterized combination function, e.g., a weighted sum, is used, such that the parameters maximize the predictive accuracy of the system.

[0046] Thus, in one embodiment the probability of revision (PR) of a nearby query to a known highly-ranked query is the behavioral similarity (BS) of the NQ to the KHRQ, i.e., number of times the nearby query has been revised (R) to the known highly-ranked query (R(NQ, KHRQ)) over the query occurrence frequency of the nearby query, each of which is determined from the log file **110** data, i.e., PR(NQ, KHRQ)=BS(NQ, KHRQ)=R(NQ, KHRQ)/QF(NQ). As a nearby query may have a record of being revised to multiple KHRQs, this calculation is made separately for each KHRQ. Once PR is determined, queries with statistically significant PR retain their classification as nearby queries, and queries with lower PR are classified as other queries (OQs). All KHRQs and NQs are stored in an index, with a pointer from each NQ to each of its respective KHRQs. The KHRQs and NQs in the index are collectively referred to as indexed queries (IQs). PR for each indexed query also is stored in the index.

[0047] **FIG. 4** shows an exemplary graphed topology of known highly-ranked queries (KHQQ), nearby queries (NQ), and other queries (OQ). As shown, queries one link away from a known highly-ranked query usually are classified as a nearby queries. However, queries that are farther from a known highly-ranked query are more likely to be classified as other queries, i.e., are likely to have only a negligible PR(KHRQ). As shown, PR decreases as distance from the KHRQ increases. In one embodiment, the path length between a known highly-ranked query and another query is directly factored into the probability measure.

[0048] Next, either as a continuation of the backend processes described above, or at runtime, the query rank reviser **108***a* measures the revision probability (RP) of a given query (GQ) to each indexed query (IQ), as measured by the similarity between the GQ and IQ. As discussed above for PR, RP can be determined based on semantic similarity, syntactic similarity, behavioral similarity, or any combination thereof. Because in some instances RP is calculated in the same manner as PR and in some instances RP uses a different calculation, RP is used herein for evaluating the revision probability of a GQ to a KHRQ, where PR is used for calculating the probability of revision of a NQ to a KHRQ. As a backend process, the RP is calculated for each query as stored in the log files **110**. As a front end process, the RP is calculated for a query as entered by a user, for example via client **118**.

[0049] In one embodiment, the RP of a given query to a NQ (RP(GQ, NQ)) is the behavioral, semantic, and syntactic similarity of the GQ to the NQ. The RP of a given query to a known highly-ranked query (RP(GQ, KHRQ)) is calculated both directly and indirectly. The direct portion uses the standard RP calculation (RP(GQ, KHRQ)). The indirect portion is calculated as the sum, over all NQs that have a pointer to the KHRQ, of the product of the RP(GQ, NQ) as defined above and the RP of the NQ to the KHRQ (RP(NQ, KHRQ)). Thus, the RP of a GQ to a KHRQ is calculated both directly and indirectly, with respect to the relationship between all NQs for the KHRQ, i.e.

$$RP(GQ, KHRQ) = RP(GQ, KHRQ) + \sum_{NQs} [RP(GQ, NQ) \times PR(NQ, KHRQ)].$$

As a result, the less likely the probability of the NQ being revised to the KHRQ, the lower the RP. An example of a situation in which the indirect aspect of RP(GQ, KHRQ) would require the above equation can be seen with reference to **FIG. 5**. For example, for the RP of GQ **505** to KHRQ **510**, the second half of the equation above would be the sum of the calculations for NQ**1515**, NQ**2520**, NQ**3525**, and NQ**4530**.

[0050] Next, the query rank reviser **108***a* calculates a revision score (RS) for each indexed query as the product of the RP for the indexed query (with respect to the given query) and the query rank for the indexed query, i.e., RS(IQ)=(GQ, IQ)×QR(IQ). The indexed queries with the highest revision scores are retrieved as alternative queries (AQ). In one embodiment, indexed queries with the top ten revision scores are retrieved. In another embodiment, the indexed queries with the top one hundred revision scores are retrieved. Once the list of alternative queries is retrieved, the alternative queries that are known highly-ranked queries are provided as candidate revisions. For the alternative queries that are nearby queries, the corresponding known highly-ranked query are provided as candidate revisions, using the pointers stored in the index.

[0051] As described above, each candidate revision can be associated with a confidence measure representing the probability that the revision is a good revision. In the case of the query rank reviser **108***a*, the revision score of the alternative query for a candidate revision is used as the confidence measure for that query.

[0052] In addition, queries that have already been revised by this or another reviser **108** can be further revised by the query rank reviser **108***a*.

[0053] An example of suggesting known highly-ranked queries using the query rank reviser **108***a* follows. A first query entered by a user is [BBQ skewers]**405**. In this example, the user is interested in information about barbeque skewers. The query rank reviser **108***a* calculates or retrieves the revision probability of the query with respect to indexed queries. For this example, and referring again to **FIG. 4**, four indexed queries are used as the indexed queries: [Britney Spears]**410**, a KHRQ, and one of its NQs, [B Spears]**420**, and [Williams-Sonoma]**430**, a KHRQ, and one of its NQs, [wooden skewers]**440**. For the NQs, the probability of each being revised to its respective KHRQ (PR) also is retrieved from the index. The revision probabilities for the indexed queries **410**-**440** are:

[0054] RP([BBQ skewers], [Britney Spears])=0.11

[0055] RP([BBQ skewers], [B Spears])=0.3 (S)×0.8 (PR)=0.24

[0056] RP([BBQ skewers], [Williams-Sonoma])=0.05

[0057] RP([BBQ skewers], [wooden skewers])=0.95 (S)×0.3 (PR)=0.285

[0058] Thus, both of the KHRQs have a fairly low revision probability (RP) with respect to [BBQ skewers]. [B Spears] also has a relatively low RP to [BBQ skewers], but has a relatively high PR to [Britney Spears]. [wooden skewers] has a high RP to [BBQ skewers], but a low PR with respect to the KHRQ [Williams-Sonoma].

[0059] Next the query rank reviser **108***a* finds the revision score (RS) for each indexed query as a function of the revision probabilities from above and the query rank for each indexed query **410-440**, i.e., RS(IQ)=S(GQ, IQ)× QR(IQ). The revision scores are calculated as follows:

[0060] RS([Britney Spears])=0.11×0.93 QR([Britney Spears])=0.1023

[0061] RS([B Spears])=0.24×0.35 QR([B Spears])=0.084

[0062] RS([Williams-Sonoma])=0.05×0.75 QR([Williams-Sonoma])=0.0375

[0063] RS([wooden skewers])=0.285×0.36 QR([wooden skewers])=0.1026

[0064] Thus, although [B Spears] is the second most similar to [BBQ skewers], it has a low query rank, and thus ends up with a low RS. In addition, [Williams-Sonoma] has a high query rank, but a RP so low that it has the lowest RS of the group. The two highest RSs in the indexed queries **410-440** are [Britney Spears], which has a very high query rank and a low RP, and [wooden skewers], which has a low-to-medium query rank, but the highest RP for the group. For this example, we will assume that the top fifty percent of revision scores are retrieved as alternative queries (AQs). Therefore, [Britney Spears] and [wooden skewers] are retrieved as alternative queries.

[0065] [Britney Spears], which is a KHRQ, is returned as a candidate revision query. In addition, [Williams-Sonoma], which is the KHRQ for alternative query [wooden skewers], also is returned as a candidate revision query. In addition, because the confidence measure for the candidate revision queries, i.e., the revision score of the associated alternative query, is used in the decision whether to provide the candidate revisions to the user, the user ultimately may see only [Williams-Sonoma]. As a result, the user ends up with a suggested query [Williams-Sonoma] that sells the item for which the user was looking (wooden skewers for barbequing).

Generating Revision Confidence Measures at Runtime

[0066] Referring now to **FIG. 1***b*, there is shown another embodiment of an information retrieval system in accordance with the present invention. In addition to the previously described elements of **FIG. 1***a*, there are a session tracker **114** and a reviser confidence estimator **112**. As discussed above, a query reviser **108** may provide a confidence measure with one or more of the revised queries that it provides to the revision server **107**. The revision server **107** uses the confidence measures to determine which of the possible revised queries to select for inclusion on the revised queries page **300**. In one embodiment, confidence measures can be derived at runtime, based at least in part on historical user activity in selecting revised queries with respect to a given original query.

[0067] In the embodiment of **FIG. 1***b*, the front-end server **102** provides the session tracker **114** with user click-through behavior, along with the original query and revised query information. The session tracker **114** maintains log files **110** that store each user query in association with which query revision links **302** were accessed by the user, the results associated with each revised query, along with various features of the original query and revised queries for modeling the quality of the revised queries. The stored information can include, for example:

[0068] For the original query:

[0069] the original query itself;

[0070] each word in original query;

[0071] length of original query;

[0072] topic cluster of the original query;

[0073] the information retrieval score for the original query; and

[0074] the number of results for the original query.

[0075] For a revised query:

[0076] the revised query itself;

[0077] each word in the revised query;

[0078] identification of the revision technique that generated it;

[0079] length of revised query;

[0080] topic cluster associated with the revised query;

[0081] information retrieval score (e.g., page rank) for top search result;

[0082] number of results found for revised query;

[0083] length of click on revised query link **302**; and

[0084] length of click on revised query results **304**.

[0085] Topic clusters for queries are identified using any suitable topic identification method. One suitable method is described in U.S. application Ser. No. 10/676,571, filed on Sep. 30, 2003, entitled "Method and Apparatus for Characterizing Documents Based on Clusters of Related Words," which is incorporated by reference.

[0086] The reviser confidence estimator **112** analyzes the log files **110** using a predictive model, e.g., a multiple, logistic regression model, to generate a set of rules based on the features of the query and the revised queries that can be used to estimate the likelihood of a revised query being a successful revision for a given query. One suitable regression model is described in U.S. application Ser. No. 10/734,584, filed Dec. 15, 2003, entitled "Large Scale Machine Learning Systems and Methods," which is incorporated by reference. The reviser confidence estimator **112** operates on the assumption that certain behaviors, e.g., a long click by a user on a revised query link **302** indicates that the user is satisfied with the revision as being an accurate representation of the user's original information need. A long click can be deemed to occur when the user stays on the clicked through page for some minimum period of time, for example a minimum of 60 seconds. From the length of the clicks on the revised query links **302**, the reviser confidence estimator **112** can train the predictive model to predict the likelihood of a long click given the various features of the revised query and the original query. Revised queries having high pre-

dicted likelihoods of a long click are considered to be better (i.e., more successful) revisions for their associated original queries.

[0087] In one embodiment for a predictive model the confidence estimator **112** selects features associated with the revised queries, collects user data, such as click data, from the log files, formulates rules using the features and user data, and adds the rules to the predictive model. In addition, the confidence estimator **112** can formulate additional rules using the user data and selectively add the additional rules to the model.

[0088] At runtime, the revision server **107** provides the reviser confidence estimator **112** with the original query, and each of the revised queries received from the various query revisers **108**. The reviser confidence estimator **112** applies the original query and revised queries to the predictive model to obtain the prediction measures, which serve as the previously mentioned confidence measures. Alternatively, each query reviser **108** can directly call the reviser confidence estimator **112** to obtain the prediction measures, and then pass these values back to the revision server **107**. Although the depicted embodiment shows the reviser confidence estimator **112** as a separate module, the revision server **107** may provide the confidence estimator functionality instead. In either case, the revision server **107** uses the confidence measures, as described above, to select and order which revised queries will be shown to the user.

[0089] In one embodiment, revision server **107** uses the confidence measures to determine whether to show query revisions at all, and if so, how prominently to place the revisions or the link thereto. To do so, the revision server **107** may use either the initial confidence measures discussed previously or the dynamically generated confidence measures discussed above. For example, if the best confidence measure falls below a threshold value, this can indicate that none of the potential candidate revisions is very good, in which case no modification is made to the original result page **200**. On the other hand, if one or more of the revised queries has a very high confidence measure above another threshold value, the revision server **107** can force the query revisions, or the link to the revised query page **300**, to be shown very prominently on the original result page **200**, for example, near the top of page and in a distinctive font, or in some other prominent position. If the confidence measures are in between the two thresholds, then a link to the revised query page **300** can be placed in a less prominent position, for example at the end of the search results page **200**, e.g., as shown for link **206**. In one embodiment, whether or where to display to the user is based in part on user dissatisfaction with the original query, based for example on zero or few results, or a low information retrieval scores.

[0090] The steps of the processes described above can performed in parallel (e.g., getting results for a query revision and calculating a confidence measure for the query revision), and/or interleaved (e.g., receiving multiple query revisions from the query revisers and constructing a sorted list of query revisions on-the-fly, rather than receiving all the query revisions and then sorting the list of query revisions). In addition, although the embodiments above are described in the context of a client/server search system, the invention can also be implemented as part of a stand-alone machine (e.g., a stand-alone PC). This could be useful, for example, in the context of a desktop search application such as Google Desktop Search.

[0091] The present invention has been described in particular detail with respect to one possible embodiment. Those of skill in the art will appreciate that the invention may be practiced in other embodiments. First, the particular naming of the components, capitalization of terms, the attributes, data structures, or any other programming or structural aspect is not mandatory or significant, and the mechanisms that implement the invention or its features may have different names, formats, or protocols. Further, the system may be implemented via a combination of hardware and software, as described, or entirely in hardware elements. Also, the particular division of functionality between the various system components described herein is merely exemplary, and not mandatory; functions performed by a single system component may instead be performed by multiple components, and functions performed by multiple components may instead be performed by a single component.

[0092] Some portions of the above description present the features of the present invention in terms of algorithms and symbolic representations of operations on information. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. These operations, while described functionally or logically, are understood to be implemented by computer programs. Furthermore, it has also proven convenient at times to refer to these arrangements of operations as modules or by functional names, without loss of generality.

[0093] Unless specifically stated otherwise as apparent from the above discussion, it is appreciated that throughout the description the described actions and processes are those of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system memories or registers or other such information storage, transmission, or display devices. A detailed description of the underlying hardware of such computer systems is not provided herein as this information is commonly known to those of skill in the art of computer engineering.

[0094] Certain aspects of the present invention include process steps and instructions described herein in the form of an algorithm. It should be noted that the process steps and instructions of the present invention could be embodied in software, firmware, or hardware, and when embodied in software, could be downloaded to reside on and be operated from different platforms used by real time network operating systems.

[0095] Certain aspects of the present invention have been described with respect to individual or singular examples; however it is understood that the operation of the present invention is not limited in this regard. Accordingly, all references to a singular element or component should be interpreted to refer to plural such components as well. Likewise, references to "a,""an," or "the" should be interpreted to include reference to pluralities, unless expressed stated otherwise. Finally, use of the term "plurality" is meant to refer to two or more entities, items of data, or the like, as

appropriate for the portion of the invention under discussion, and does cover an infinite or otherwise excessive number of items.

[0096] The present invention also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored on a computer readable medium that can be accessed by the computer. Such a computer program may be stored in a computer readable storage medium, such as, but not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus. Those of skill in the art of integrated circuit design and video codecs appreciate that the invention can be readily fabricated in various types of integrated circuits based on the above functional and structural descriptions, including application specific integrated circuits (ASICs). In addition, the present invention may be incorporated into various types of video coding devices.

[0097] The algorithms and operations presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may also be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will be apparent to those of skill in the art, along with equivalent variations. In addition, the present invention is not described with reference to any particular programming language. It is appreciated that a variety of programming languages may be used to implement the teachings of the present invention as described herein, and any references to specific languages are provided for disclosure of enablement and best mode of the present invention.

[0098] Finally, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention.

We claim:

1. A method for automatically suggesting known highly-ranked queries in response to a first query, comprising:

calculating a revision score for an indexed query as a function of a revision probability for the first query with respect to the indexed query and a query rank for the indexed query;

responsive to the revision score, selectively retrieving the indexed query as an alternative query to the first query; and

responsive to the alternative query being a known highly-ranked query, returning the alternative query as a candidate revision query.

2. The method of claim 1, further comprising:

responsive to the alternative query having a statistically significant probability of revising to a known highly-ranked query, returning the known highly-ranked query as a candidate revision query.

3. The method of claim 1, wherein the first query is a query revision.

4. The method of claim 1, wherein the function is the product of the revision probability for the first query with respect to the indexed query and a query rank for the indexed query.

5. The method of claim 1, further comprising identifying a query as the known highly-ranked query.

6. The method of claim 5, further comprising:

calculating a query occurrence frequency for the query;

calculating a user satisfaction score for the query; and

computing a rank for the query as a function of the query occurrence frequency and user satisfaction score.

7. The method of claim 6, wherein the user satisfaction score is determined by user click behavior data estimating the length of clicks on search results.

8. The method of claim 6, wherein the user satisfaction score is determined by an inverse revision frequency.

9. The method of claim 8, wherein the inverse revision frequency is the inverse of a number of times the query is revised divided by the query occurrence frequency of the query.

10. The method of claim 1, further comprising creating an index of queries comprising the indexed query.

11. The method of claim 10, wherein the index includes pointers from each of the queries in the index to one or more known highly-ranked queries.

12. The method of claim 1, wherein the revision probability comprises behavioral similarity of the indexed query with respect to the first query.

13. The method of claim 1, wherein the revision probability comprises semantic similarity of the indexed query with respect to the first query.

14. The method of claim 1, wherein the revision probability comprises syntactical similarity of the indexed query with respect to the first query.

15. The method of claim 1, further comprising:

logging query data generated from user sessions; and

using the query data to generate the index of queries.

16. The method of claim 1, further comprising:

ranking the candidate revision query using the revision score for the candidate revision query as a confidence measure.

17. The method of claim 1, further comprising:

providing the candidate revision query as a suggested revision for the first query.

18. The method of claim 17, wherein the suggested revision is displayed to a user in a location dependent upon a relative strength of the confidence measure.

19. A method for automatically suggesting known highly-ranked queries in response to a first query, comprising:

logging query data generated from user sessions;

creating an index of queries during the user session;

calculating a revision score for an indexed query as a function of a revision probability for the first query with respect to the indexed query and a query rank for

the indexed query, wherein the revision probability comprises the similarity of the indexed query with respect to the first query;

responsive to the revision score, selectively retrieving the indexed query as an alternative query to the first query;

responsive to the alternative query being a known highly-ranked query, returning the alternative query as a candidate revision query, wherein identifying the known highly-ranked query comprises:

calculating a query occurrence frequency for a query;

calculating a user satisfaction score for the query, wherein the user satisfaction score is determined by user click behavior data estimating the length of clicks on search results; and

computing a rank for the query as a product of the query occurrence frequency and the user satisfaction score;

responsive to the alternative query having a statistically significant probability of revising to a known highly-ranked query, returning the known highly-ranked query as a candidate revision query;

ranking the candidate revision query using the revision score for the candidate revision query as a confidence measure; and

providing the candidate revision query as a suggested revision for the first query, wherein the suggested revision is displayed to a user in a location dependent upon a relative strength of the confidence measure.

20. A method of identifying a query as a known highly-ranked query, comprising:

calculating a query occurrence frequency for the query;

calculating a user satisfaction score for the query; and

computing a query rank for the query using the query occurrence frequency and the user satisfaction score.

21. The method of claim 20, further comprising:

comparing the query rank of the query to other query ranks associated with the other queries to determine whether the query rank is greater than the other query ranks.

22. The method of claim 20, wherein the query occurrence frequency is calculated as the frequency of the query per unit time.

23. The method of claim 20, wherein the user satisfaction score is based upon an inverse revision frequency.

24. The method of claim 23, wherein the inverse revision frequency is the inverse of a number of times the query is revised divided by the query occurrence frequency of the query.

25. The method of claim 20, wherein the user satisfaction score is a function of a quality score.

26. The method of claim 25, wherein the quality score for the query is estimated from user behavior data estimating the length of clicks on search results.

27. The method of claim 20, wherein the query rank is a function of the query occurrence frequency and the user satisfaction score.

28. The method of claim 20, wherein the known highly-ranked query is returned as a candidate revision query.

29. A computer program product for automatically suggesting known highly-ranked queries in response to a first query, the computer program product comprising:

a computer-readable medium; and

computer program code, coded on the medium, for:

calculating a revision score for an indexed query as a function of a revision probability for the first query with respect to the indexed query and a query rank for the indexed query;

responsive to the revision score, selectively retrieving the indexed query as an alternative query to the first query; and

responsive to the alternative query being a known highly-ranked query, returning the alternative query as a candidate revision query.

30. The computer program product of claim 29, further comprising:

computer program code, coded on the medium, for:

returning the known highly-ranked query as a candidate revision query responsive to the alternative query having a statistically significant probability of revising to a known highly-ranked query.

31. The computer program product of claim 29, further comprising:

computer program code, coded on the medium, for:

ranking the candidate revision query using the revision score for the candidate revision query as a confidence measure.

32. A computer program product for identifying a query as a known highly-ranked query, the computer program product comprising:

a computer-readable medium; and

computer program code, coded on the medium, for:

calculating a query occurrence frequency for the query;

calculating a user satisfaction score for the query; and

computing a query rank using the query occurrence frequency and the user satisfaction score.

33. The computer program product of claim 32, wherein the user satisfaction score is a function of a quality score.

34. The computer program product of claim 32, wherein the quality score for the query is estimated from user behavior data estimating the length of clicks on search results.

35. The computer program product of claim 32, wherein the query rank is a function of the query occurrence frequency and the user satisfaction score.

36. A system for providing revised queries for a query as a known highly-ranked query, the system comprising:

means for calculating a revision score for an indexed query as a function of a revision probability for the first query with respect to the indexed query and a query rank for the indexed query;

means for responsive to the revision score, selectively retrieving the indexed query as an alternative query to the first query; and

means for responsive to the alternative query being a known highly-ranked query, returning the alternative query as a candidate revision query.

37. The system of claim 36, further comprising:

means for returning the known highly-ranked query as a candidate revision query responsive to the alternative query having a statistically significant probability of revising to a known highly-ranked query.

38. The system of claim 36, further comprising:

means for ranking the candidate revision query using the revision score for the candidate revision query as a confidence measure.

39. A system for providing revised queries for identifying a query as a known highly-ranked query, the system comprising:

means for calculating a query occurrence frequency for the query;

calculating a user satisfaction score for the query; and

computing a query rank using the query occurrence frequency and the user satisfaction score.

40. The system of claim 39, wherein the user satisfaction score is a function of a quality score.

41. The system of claim 39, wherein the quality score for the query is estimated from user behavior data estimating the length of clicks on search results.

42. The system of claim 39, wherein the query rank is a function of the query occurrence frequency and the user satisfaction score.

* * * * *