(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 March 2009 (12.03.2009)

PCT

(10) International Publication Number
WO 2009/033098 A1

(51) International Patent Classification:
G06F 7/00 (2006.01)

(21) International Application Number:
PCT/US2008/075492

(22) International Filing Date:
5 September 2008 (05.09.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/851,962 7 September 2007 (07.09.2007) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 11/851,962 (CIP)
Filed on 7 September 2007 (07.09.2007)

(71) Applicant *(for all designated States except US)*: **GOOGLE INC.** [US/US]; 1600 Amphitheatre Parkway, Mountain View, CA 94043 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **PATTERSON, Anna, L.** [US/US]; 1127 Thorntree Court, San Jose, CA 95120 (US).

(74) Agent: **LIVESAY, Margo**; Brake Hughes Bellermann LLP, c/o Intellevate, PO Box 52050, Minneapolis, MN 55402 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US (patent), UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH,

[Continued on next page]

(54) Title: INTEGRATING EXTERNAL RELATED PHRASE INFORMATION INTO A PHRASE-BASED INDEXING INFORMATION RETRIEVAL SYSTEM

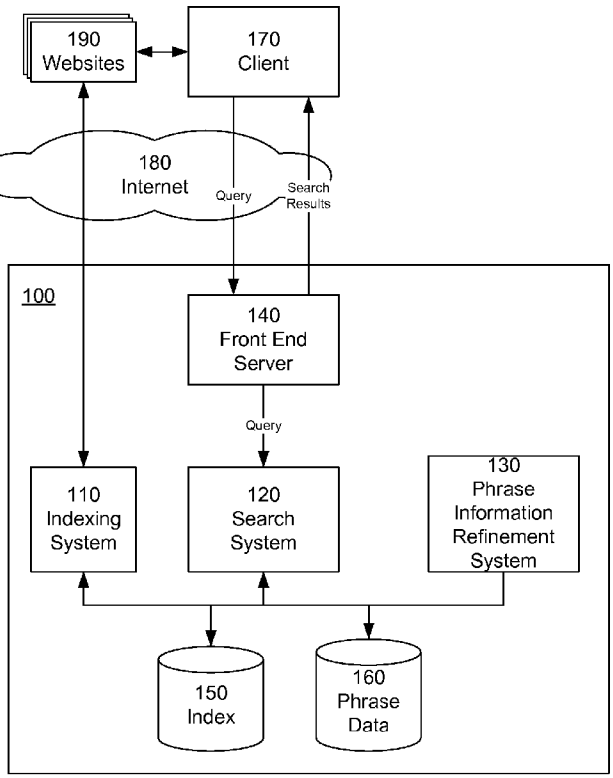

FIG. 1

(57) Abstract: An information retrieval system uses phrases to index, retrieve, organize and describe documents, analyzing documents and storing the results of the analysis as phrase data. Phrases are identified that predict the presence of other phrases in documents. Documents are the indexed according to their included phrases. Related phrases and phrase extensions are also identified. Changes to existing phrase data about a document collection submitted by a user is captured and analyzed, and the existing phrase data is updated to reflect the additional knowledge gained through the analysis.

WO 2009/033098 A1

GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

INTEGRATING EXTERNAL RELATED PHRASE INFORMATION INTO A PHRASE-BASED INDEXING INFORMATION RETRIEVAL SYSTEM

Inventor: Anna L. Patterson

Priority Claim

This application claims the benefit of priority based on United States Application No. 11/851,962, filed on September 7, 2007, entitled, "Integrating External Related Phrase Information into a Phrase-Based Indexing Information Retrieval System," the disclosure of which is hereby incorporated by reference herein.

Cross Reference to Related Applications

[0001] The application is related to the following co-pending U.S. applications: "Phrase Identification in an Information Retrieval System," Application No. 10/900,021, filed on July 26, 2004; "Phrase-Based Indexing in an Information Retrieval System," Application No. 10/900,055, filed on July 26, 2004; "Phrase-Based Searching in an Information Retrieval System," Application No. 10/900,041, filed on July 26, 2004; "Phrase-Based Personalization of Searches in an Information Retrieval System," Application No. 10/900,039, filed on July 26, 2004; "Automatic Taxonomy Generation in Search Results Using Phrases," Application No. 10/900,259, filed on July 26, 2004; and "Phrase-Based Detection of Duplicate Documents in an Information Retrieval System," Application No. 10/900,012, filed on July 26, 2004, all of which are co-owned, and incorporated by reference herein.

Field of the Invention

[0002] The present invention relates to an information retrieval system for indexing, searching, and classifying documents in a large scale corpus, such as the Internet.

Background of the Invention

[0003] Information retrieval systems, generally called search engines, are now an essential tool for finding information in large scale, diverse, and growing corpuses such

as the Internet. Generally, search engines create an index that relates documents (or "pages") to the individual words present in each document. A document is retrieved in response to a query containing a number of query terms, typically based on having some number of query terms present in the document. The retrieved documents are then ranked according to other statistical measures, such as frequency of occurrence of the query terms, host domain, link analysis, and the like. The retrieved documents are then presented to the user, typically in their ranked order, and without any further grouping or imposed hierarchy. In some cases, a selected portion of a text of a document is presented to provide the user with a glimpse of the document's content.

[0004] Direct "Boolean" matching of query terms has well known limitations, and in particular does not identify documents that do not have the query terms, but have related words. For example, in a typical Boolean system, a search on "Australian Shepherds" would not return documents about other herding dogs such as Border Collies that do not have the exact query terms. Rather, such a system is likely to also retrieve and highly rank documents that are about Australia (and have nothing to do with dogs), and documents about "shepherds" generally.

[0005] The problem here is that conventional systems index documents based on individual terms, than on concepts. Concepts are often expressed in phrases, such as "Australian Shepherd," "President of the United States," or "Sundance Film Festival". At best, some prior systems will index documents with respect to a predetermined and very limited set of 'known' phrases, which are typically selected by a human operator. Indexing of phrases is typically avoided because of the perceived computational and memory requirements to identify all possible phrases of say three, four, or five or more words. For example, on the assumption that any five words could constitute a phrase,

and a large corpus would have at least 200,000 unique terms, there would approximately 3.2×10^{26} possible phrases, clearly more than any existing system could store in memory or otherwise programmatically manipulate. A further problem is that phrases continually enter and leave the lexicon in terms of their usage, much more frequently than new individual words are invented. New phrases are always being generated, from sources such technology, arts, world events, and law. Other phrases will decline in usage over time.

[0006] Some existing information retrieval systems attempt to provide retrieval of concepts by using co-occurrence patterns of individual words. In these systems a search on one word, such as "President" will also retrieve documents that have other words that frequently appear with "President", such as "White" and "House." While this approach may produce search results having documents that are conceptually related at the level of individual words, it does not typically capture topical relationships that inhere between co-occurring phrases.

[0007] Accordingly, there is a need for an information retrieval system and methodology that can comprehensively identify phrases in a large scale corpus, index documents according to phrases, search and rank documents in accordance with their phrases. Additionally, there is a need in such a system to allow users to provide additional phrase information to the system and to capture and integrate the resulting semantic knowledge.

Summary of the Invention

[0008] An information retrieval system and methodology uses phrases to index, search, rank, and describe documents in the document collection. The system is

adapted to identify phrases that have sufficiently frequent and/or distinguished usage in the document collection to indicate that they are "valid" or "good" phrases. In this manner multiple word phrases, for example phrases of four, five, or more terms, can be identified. This avoids the problem of having to identify and index every possible phrase resulting from all of the possible sequences of a given number of words.

[0009] The system is further adapted to identify phrases that are related to each other, based on a phrase's ability to predict the presence of other phrases in a document. More specifically, a prediction measure is used that relates the actual co-occurrence rate of two phrases to an expected co-occurrence rate of the two phrases. Information gain, as the ratio of actual co-occurrence rate to expected co-occurrence rate, is one such prediction measure. Two phrases are related where the prediction measure exceeds a predetermined threshold. In that case, the second phrase has significant information gain with respect to the first phrase. Semantically, related phrases will be those that are commonly used to discuss or describe a given topic or concept, such as "President of the United States" and "White House." For a given phrase, the related phrases can be ordered according to their relevance or significance based on their respective prediction measures.

[0010] An information retrieval system indexes documents in the document collection by the valid or good phrases. For each phrase, a posting list identifies the documents that contain the phrase. In addition, for a given phrase, a second list, vector, or other structure is used to store data indicating which of the related phrases of the given phrase are also present in each document containing the given phrase. In this manner, the system can readily identify not only which documents contain which phrases in response to a search query, but which documents also contain phrases that

are related to query phrases, and thus more likely to be specifically about the topics or concepts expressed in the query phrases.

[0011] The use of phrases and related phrases further provides for the creation and use of clusters of related phrases, which represent semantically meaningful groupings of phrases. Clusters are identified from related phrases that have very high prediction measure between all of the phrases in the cluster. Clusters can be used to organize the results of a search, including selecting which documents to include in the search results and their order, as well as eliminating documents from the search results.

[0012] Websites typically have anywhere from a few pages to potentially hundreds or thousands of pages. Thus, phrase information generated by the information retrieval system can be used to determine a list of top phrases for each website, such as the most representative phrases for the website. This can be done by examining the related phrase information for the phrases that appear in documents on the website. Further, phrase information may be later supplemented and refined by capturing changes made to the top phrase list by administrators or other authorized users and integrating the resulting semantic knowledge into the phrase information already contained within the system. An administrator can associate additional related phrases with any of the top phrases for the website. The related phrase information for the top phrases for which additional related phrases have been received is then updated to include information pertaining to the additional related phrases, and the additional related phrases are also updated to include information from the top phrases. This operates to treat the additional phrases as if they were present in the website. In addition, the additional related phrases can be updated to use the related phrase information for the top phrases.

[0013] The present invention has further embodiments in system and software architectures, computer program products and computer implemented methods, and computer generated user interfaces and presentations.

[0014] The foregoing are just some of the features of an information retrieval system and methodology based on phrases. Those of skill in the art of information retrieval will appreciate the flexibility of generality of the phrase information allows for a large variety of uses and applications in indexing, document annotation, searching, ranking, and other areas of document analysis and processing.

Brief Description of the Drawings

[0015] FIG. 1 is block diagram of the software architecture of one embodiment of the present invention.

[0016] FIG. 2 illustrates a method of identifying phrases in documents.

[0017] FIG. 3 illustrates a document with a phrase window and a secondary window.

[0018] FIG. 4 illustrates a method of identifying related phrases.

[0019] FIG. 5 illustrates a method of indexing documents for related phrases.

[0020] FIG. 6 illustrates a method of retrieving documents based on phrases.

[0021] FIGS. 7a and 7b illustrate relationships between referencing and referenced documents.

[0022] FIG. 8 illustrates a method of obtaining and integrating phrase information input from users.

[0023] FIG. 9 illustrates a sample user interface for displaying top phrases and allowing users to input changes.

[0024] The figures depict a preferred embodiment 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 structures and methods illustrated herein may be employed without departing from the principles of the invention described herein.

Detailed Description of the Invention

I. System Overview

[0025] Referring now to FIG. 1, there is shown the software architecture of an embodiment of a search system 100 in accordance with one embodiment of present invention. In this embodiment, the system includes a indexing system 110, a search system 120, a presentation system 130, and a front end server 140.

[0026] The indexing system 110 is responsible for identifying phrases in documents, and indexing documents according to their phrases, by accessing various websites 190 and other document collections. The front end server 140 receives queries from a user of a client 170, and provides those queries to the search system 120. The search system 120 is responsible for searching for documents relevant to the search query (search results), including identifying any phrases in the search query, and then ranking the documents in the search results using the presence of phrases to influence the ranking order. The search system 120 provides the search results to the presentation system 130. The presentation system 130 is responsible for modifying the search results including removing near duplicate documents, and generating topical descriptions of documents, and providing the modified search results back to the front end server 140, which provides the results to the client 170. The system 100 further includes an index

150 that stores the indexing information pertaining to documents, and a phrase data store 160 that stores phrases, and related statistical information.

[0027] In the context of this application, "documents" are understood to be any type of media that can be indexed and retrieved by a search engine, including web documents, images, multimedia files, text documents, PDFs or other image formatted files, and so forth. A document may have one or more pages, partitions, segments or other components, as appropriate to its content and type. Equivalently a document may be referred to as a "page," as commonly used to refer to documents on the Internet. No limitation as to the scope of the invention is implied by the use of the generic term "documents." The search system 100 operates over a large corpus of documents, such as the Internet and World Wide Web, but can likewise be used in more limited collections, such as for the document collections of a library or private enterprises. In either context, it will be appreciated that the documents are typically distributed across many different computer systems and sites. Without loss of generality then, the documents generally, regardless of format or location (e.g., which website or database) will be collectively referred to as a corpus or document collection. Each document has an associated identifier that uniquely identifies the document; the identifier is preferably a URL, but other types of identifiers (e.g., document numbers) may be used as well. In this disclosure, the use of URLs to identify documents is assumed.

II. Indexing System

[0028] In one embodiment, the indexing system 110 provides three primary functional operations: 1) identification of phrases and related phrases, 2) indexing of documents with respect to phrases, and 3) generation and maintenance of a phrase-based taxonomy. Those of skill in the art will appreciate that the indexing system 110

will perform other functions as well in support of conventional indexing functions, and thus these other operations are not further described herein. The indexing system 110 operates on an index 150 and data repository 160 of phrase data. These data repositories are further described below.

1. Phrase Identification

[0029] The phrase identification operation of the indexing system 110 identifies "good" and "bad" phrases in the document collection that are useful to indexing and searching documents. In one aspect, good phrases are phrases that tend to occur in more than certain percentage of documents in the document collection, and/or are indicated as having a distinguished appearance in such documents, such as delimited by markup tags or other morphological, format, or grammatical markers. Another aspect of good phrases is that they are predictive of other good phrases, and are not merely sequences of words that appear in the lexicon. For example, the phrase "President of the United States" is a phrase that predicts other phrases such as "George Bush" and "Bill Clinton." However, other phrases are not predictive, such as "fell down the stairs" or "top of the morning," "out of the blue," since idioms and colloquisms like these tend to appear with many other different and unrelated phrases. Thus, the phrase identification phase determines which phrases are good phrases and which are bad (i.e., lacking in predictive power).

[0030] Referring to now FIG. 2, the phrase identification process has the following functional stages:

[0031] 200: Collect possible and good phrases, along with frequency and co-occurrence statistics of the phrases.

[0032] 202: Classify possible phrases to either good or bad phrases based on frequency statistics.

[0033] 204: Prune good phrase list based on a predictive measure derived from the co-occurrence statistics.

[0034] Each of these stages will now be described in further detail.

[0035] The first stage 200 is a process by which the indexing system 110 crawls a set of documents in the document collection, making repeated partitions of the document collection over time. One partition is processed per pass. The number of documents crawled per pass can vary, and is preferably about 1,000,000 per partition. It is preferred that only previously uncrawled documents are processed in each partition, until all documents have been processed, or some other termination criteria is met. In practice, the crawling continues as new documents are being continually added to the document collection. The following steps are taken by the indexing system 110 for each document that is crawled.

[0036] Traverse the words of the document with a phrase window length of n, where n is a desired maximum phrase length. The length of the window will typically be at least 2, and preferably 4 or 5 terms (words). Preferably phrases include all words in the phrase window, including what would otherwise be characterized as stop words, such as "a", "the," and so forth. A phrase window may be terminated by an end of line, a paragraph return, a markup tag, or other indicia of a change in content or format.

[0037] FIG. 3 illustrates a portion of a document 300 during a traversal, showing the phrase window 302 starting at the word "stock" and extending 5 words to the right. The first word in the window 302 is candidate phrase *i*, and the each of the sequences *i*+1, *i*+2, *i*+3, *i*+4, and *i*+5 is likewise a candidate phrase. Thus, in this example, the

candidate phrases are: "stock", "stock dogs", "stock dogs for", "stock dogs for the", "stock dogs for the Basque", and "stock dogs for the Basque shepherds".

[0038] In each phrase window 302, each candidate phrase is checked in turn to determine if it is already present in the good phrase list 208 or the possible phrase list 206. If the candidate phrase is not present in either the good phrase list 208 or the possible phrase list 206, then the candidate has already been determined to be "bad" and is skipped.

[0039] If the candidate phrase is in the good phrase list 208, as entry g_j, then the index 150 entry for phrase g_j is updated to include the document (e.g., its URL or other document identifier), to indicate that this candidate phrase g_j appears in the current document. An entry in the index 150 for a phrase g_j (or a term) is referred to as the posting list of the phrase g_j. The posting list includes a list of documents d (by their document identifiers, e.g. a document number, or alternatively a URL) in which the phrase occurs.

[0040] In addition, the co-occurrence matrix 212 is updated, as further explained below. In the very first pass, the good and bad lists will be empty, and thus, most phrases will tend to be added to the possible phrase list 206.

[0041] If the candidate phrase is not in the good phrase list 208 then it is added to the possible phrase list 206, unless it is already present therein. Each entry p on the possible phrase list 206 has three associated counts:

[0042] P(p): Number of documents on which the possible phrase appears;

[0043] S(p): Number of all instances of the possible phrase; and

[0044] M(p): Number of interesting instances of the possible phrase. An instance of a possible phrase is "interesting" where the possible phrase is distinguished from

neighboring content in the document by grammatical or format markers, for example by being in boldface, or underline, or as anchor text in a hyperlink, or in quotation marks. These (and other) distinguishing appearances are indicated by various HTML markup language tags and grammatical markers. These statistics are maintained for a phrase when it is placed on the good phrase list 208.

[0045] In addition the various lists, a co-occurrence matrix 212 (G) for the good phrases is maintained. The matrix G has a dimension of m x m, where m is the number of good phrases. Each entry G(j, k) in the matrix represents a pair of good phrases (g_j, g_k). The co-occurrence matrix 212 logically (though not necessarily physically) maintains three separate counts for each pair (g_j, g_k) of good phrases with respect to a secondary window 304 that is centered at the current word i, and extends +/- h words. In one embodiment, such as illustrated in FIG. 3, the secondary window 304 is 30 words. The co-occurrence matrix 212 thus maintains:

[0046] R(j,k): Raw Co-occurrence count. The number of times that phrase g_j appears in a secondary window 304 with phrase g_k;

[0047] D(j,k): Disjunctive Interesting count. The number of times that either phrase g_j or phrase g_k appears as distinguished text in a secondary window; and

[0048] C(j,k): Conjunctive Interesting count: the number of times that both g_j and phrase g_k appear as distinguished text in a secondary window. The use of the conjunctive interesting count is particularly beneficial to avoid the circumstance where a phrase (e.g., a copyright notice) appears frequently in sidebars, footers, or headers, and thus is not actually predictive of other text.

[0049] Referring to the example of FIG. 3, assume that the "stock dogs" is on the good phrase list 208, as well as the phrases "Australian Shepherd" and "Australian

Shepard Club of America". Both of these latter phrases appear within the secondary window 304 around the current phrase "stock dogs". However, the phrase "Australian Shepherd Club of America" appears as anchor text for a hyperlink (indicated by the underline) to website. Thus the raw co-occurrence count for the pair {"stock dogs", "Australian Shepherd"} is incremented, and the raw occurrence count and the disjunctive interesting count for {"stock dogs", "Australian Shepherd Club of America"} are both incremented because the latter appears as distinguished text.

[0050] The process of traversing each document with both the sequence window 302 and the secondary window 304, is repeated for each document in the partition.

[0051] Once the documents in the partition have been traversed, the next stage of the indexing operation is to update 202 the good phrase list 208 from the possible phrase list 206. A possible phrase p on the possible phrase list 206 is moved to the good phrase list 208 if the frequency of appearance of the phrase and the number of documents that the phrase appears in indicates that it has sufficient usage as semantically meaningful phrase.

[0052] In one embodiment, this is tested as follows. A possible phrase p is removed from the possible phrase list 206 and placed on the good phrase list 208 if:

[0053] a) $P(p) > 10$ and $S(p) > 20$ (the number of documents containing phrase p is more than 10, and the number of occurrences of phrase p is more then 20); or

[0054] b) $M(p) > 5$ (the number of interesting instances of phrase p is more than 5) .

[0055] These thresholds are scaled by the number of documents in the partition; for example if 2,000,000 documents are crawled in a partition, then the thresholds are

approximately doubled. Of course, those of skill in the art will appreciate that the specific values of the thresholds, or the logic of testing them, can be varied as desired.

[0056] If a phrase *p* does not qualify for the good phrase list 208, then it is checked for qualification for being a bad phrase. A phrase *p* is a bad phrase if:

[0057] a) number of documents containing phrase, $P(p) < 2$; and

[0058] b) number of interesting instances of phrase, $M(p) = 0$.

[0059] These conditions indicate that the phrase is both infrequent, and not used as indicative of significant content and again these thresholds may be scaled per number of documents in the partition.

[0060] It should be noted that the good phrase list 208 will naturally include individual words as phrases, in addition to multi-word phrases, as described above. This is because each the first word in the phrase window 302 is always a candidate phrase, and the appropriate instance counts will be accumulated. Thus, the indexing system 110 can automatically index both individual words (i.e., phrases with a single word) and multiple word phrases. The good phrase list 208 will also be considerably shorter than the theoretical maximum based on all possible combinations of m phrases. In typical embodiment, the good phrase list 208 will include about 6.5×10^5 phrases. A list of bad phrases is not necessary to store, as the system need only keep track of possible and good phrases.

[0061] By the final pass through the document collection, the list of possible phrases will be relatively short, due to the expected distribution of the use of phrases in a large corpus. Thus, if say by the 10^{th} pass (e.g., 10,000,000 documents), a phrase appears for the very first time, it is very unlikely to be a good phrase at that time. It may be new phrase just coming into usage, and thus during subsequent crawls becomes

increasingly common. In that case, its respective counts will increases and may ultimately satisfy the thresholds for being a good phrase.

[0062] The third stage of the indexing operation is to prune 204 the good phrase list 208 using a predictive measure derived from the co-occurrence matrix 212. Without pruning, the good phrase list 208 is likely to include many phrases that while legitimately appearing in the lexicon, themselves do not sufficiently predict the presence of other phrases, or themselves are subsequences of longer phrases. Removing these weak good phrases results in a very robust likely of good phrases. To identify good phrases, a predictive measure is used which expresses the increased likelihood of one phrase appearing in a document given the presence of another phrase. This is done, in one embodiment, as follows:

[0063] As noted above, the co-occurrence matrix 212 is an $m \times m$ matrix of storing data associated with the good phrases. Each row j in the matrix represents a good phrase g_j and each column k represented a good phrase g_k. For each good phrase g_j, an expected value $E(g_j)$ is computed. The expected value E is the percentage of documents in the collection expected to contain g_j. This is computed, for example, as the ratio of the number of documents containing g_j to the total number T of documents in the collection that have been crawled: P(j)/T.

[0064] As noted above, the number of documents containing g_j is updated each time g_j appears in a document. The value for $E(g_j)$ can be updated each time the counts for g_j are incremented, or during this third stage.

[0065] Next, for each other good phrase g_k (e.g., the columns of the matrix), it is determined whether g_j predicts g_k. A predictive measure for g_j is determined as follows:

[0066] i) compute the expected value $E(g_k)$. The expected co-occurrence rate E(j,k) of g_j and g_k, if they were unrelated phrases is then $E(g_j)*E(g_k)$;

[0067] ii) compute the actual co-occurrence rate A(j,k) of g_j and g_k. This is the raw co-occurrence count R(j, k) divided by T, the total number of documents;

[0068] iii) g_j is said to predict g_k where the actual co-occurrence rate A(j,k) exceeds the expected co-occurrence rate E(j,k) by a threshold amount.

[0069] In one embodiment, the predictive measure is information gain. Thus, a phrase g_j predicts another phrase g_k when the information gain I of g_k in the presence of g_j exceeds a threshold. In one embodiment, this is computed as follows:

[0070] I(j,k) = A(j,k)/E(j,k)

[0071] And good phrase g_j predicts good phrase g_k where:

[0072] I(j,k) > Information Gain threshold.

[0073] In one embodiment, the information gain threshold is 1.5, but is preferably between 1.1 and 1.7. Raising the threshold over 1.0 serves to reduce the possibility that two otherwise unrelated phrases co-occur more than randomly predicted.

[0074] As noted the computation of information gain is repeated for each column *k* of the matrix G with respect to a given row *j*. Once a row is complete, if the information gain for none of the good phrases g_k exceeds the information gain threshold, then this means that phrase g_j does not predict any other good phrase. In that case, g_j is removed from the good phrase list 208, essentially becoming a bad phrase. Note that the column j for the phrase g_j is not removed, as this phrase itself may be predicted by other good phrases.

[0075] This step is concluded when all rows of the co-occurrence matrix 212 have been evaluated.

[0076] The final step of this stage is to prune the good phrase list 208 to remove incomplete phrases. An incomplete phrase is a phrase that only predicts its phrase extensions, and which starts at the left most side of the phrase (i.e., the beginning of the phrase). The "phrase extension" of phrase p is a super-sequence that begins with phrase p. For example, the phrase "President of" predicts "President of the United States", "President of Mexico", "President of AT&T", etc. All of these latter phrases are phrase extensions of the phrase "President of" since they begin with "President of" and are super-sequences thereof.

[0077] Accordingly, each phrase g_j remaining on the good phrase list 208 will predict some number of other phrases, based on the information gain threshold previously discussed. Now, for each phrase g_j the indexing system 110 performs a string match with each of the phrases g_k that is predicts. The string match tests whether each predicted phrase g_k is a phrase extension of the phrase g_j. If all of the predicted phrases g_k are phrase extensions of phrase g_j, then phrase g_j is incomplete, and is removed from the good phrase list 208, and added to an incomplete phrase list 216. Thus, if there is at least one phrase g_k that is not an extension of g_j, then g_j is complete, and maintained in the good phrase list 208. For example then, "President of the United" is an incomplete phrase because the only other phrase that it predicts is "President of the United States" which is an extension of the phrase.

[0078] The incomplete phrase list 216 itself is very useful during actual searching. When a search query is received, it can be compared against the incomplete phase list 216. If the query (or a portion thereof) matches an entry in the list, then the

search system 120 can lookup the most likely phrase extensions of the incomplete phrase (the phrase extension having the highest information gain given the incomplete phrase), and suggest this phrase extension to the user, or automatically search on the phrase extension. For example, if the search query is "President of the United," the search system 120 can automatically suggest to the user "President of the United States" as the search query.

[0079] After the last stage of the indexing process is completed, the good phrase list 208 will contain a large number of good phrases that have been discovered in the corpus. Each of these good phrases will predict at least one other phrase that is not a phrase extension of it. That is, each good phrase is used with sufficient frequency and independence to represent meaningful concepts or ideas expressed in the corpus. Unlike existing systems which use predetermined or hand selected phrases, the good phrase list reflects phrases that actual are being used in the corpus. Further, since the above process of crawling and indexing is repeated periodically as new documents are added to the document collection, the indexing system 110 automatically detects new phrases as they enter the lexicon.

2. Identification of Related Phrases and Clusters of Related Phrases

[0080] Referring to FIG. 4, the related phrase identification process includes the following functional operations.

[0081] 400: Identify related phrases having a high information gain value.

[0082] 402: Identify clusters of related phrases.

[0083] 404: Store cluster bit vector and cluster number.

[0084] Each of these operations is now described in detail.

[0085] First, recall that the co-occurrence matrix 212 contains good phrases g_j, each of which predicts at least one other good phrase g_k with an information gain greater than the information gain threshold. To identify 400 related phrases then, for each pair of good phrases (g_j, g_k) the information gain is compared with a Related Phrase threshold, e.g., 100. That is, g_j and g_k are related phrases where:

[0086] $I(g_j, g_k) > 100$.

[0087] This high threshold is used to identify the co-occurrences of good phrases that are well beyond the statistically expected rates. Statistically, it means that phrases g_j and g_k co-occur 100 times more than the expected co-occurrence rate. For example, given the phrase "Monica Lewinsky" in a document, the phrase "Bill Clinton" is a 100 times more likely to appear in the same document, then the phrase "Bill Clinton" is likely to appear on any randomly selected document. Another way of saying this is that the accuracy of the predication is 99.999% because the occurrence rate is 100:1.

[0088] Accordingly, any entry (g_j, g_k) that is less the Related Phrase threshold is zeroed out, indicating that the phrases g_j, g_k are not related. Any remaining entries in the co-occurrence matrix 212 now indicate all related phrases.

[0089] The columns g_k in each row g_j of the co-occurrence matrix 212 are then sorted by the information gain values $I(g_j, g_k)$, so that the related phrase g_k with the highest information gain is listed first. This sorting thus identifies for a given phrase g_j, which other phrases are most likely related in terms of information gain.

[0090] The next step is to determine 402 which related phrases together form a cluster of related phrases. A cluster is a set of related phrases in which each phrase has high information gain with respect to at least one other phrase. In one embodiment, clusters are identified as follows.

[0091] In each row g_j of the matrix, there will be one or more other phrases that are related to phrase g_j. This set is related phrase set R_j, where $R=\{g_k, g_l, \ldots g_m\}$.

[0092] For each related phrase *m* in R_j, the indexing system 110 determines if each of the other related phrases in R is also related to g_j. Thus, if $I(g_k, g_l)$ is also non-zero, then g_j, g_k, and g_l are part of a cluster. This cluster test is repeated for each pair (g_l, g_m) in R.

[0093] For example, assume the good phrase "Bill Clinton" is related to the phrases "President", "Monica Lewinsky", because the information gain of each of these phrases with respect to "Bill Clinton" exceeds the Related Phrase threshold. Further assume that the phrase "Monica Lewinsky" is related to the phrase "purse designer". These phrases then form the set R. To determine the clusters, the indexing system 110 evaluates the information gain of each of these phrases to the others by determining their corresponding information gains. Thus, the indexing system 110 determines the information gain I("President", "Monica Lewinsky"), I("President", "purse designer"), and so forth, for all pairs in R. In this example, "Bill Clinton," "President", and "Monica Lewinsky" form a one cluster, "Bill Clinton," and "President" form a second cluster, and "Monica Lewinsky" and "purse designer" form a third cluster, and "Monica Lewinsky", "Bill Clinton," and "purse designer" form a fourth cluster. This is because while "Bill Clinton" does not predict "purse designer" with sufficient information gain, "Monica Lewinsky" does predict both of these phrases.

[0094] To record 404 the cluster information, each cluster is assigned a unique cluster number (cluster ID). This information is then recorded in conjunction with each good phrase g_j.

[0100] In one embodiment, the cluster number is determined by a cluster bit vector that also indicates the orthogonality relationships between the phrases. The cluster bit vector is a sequence of bits of length n, the number of good phrases in the good phrase list 208. For a given good phrase g_j, the bit positions correspond to the sorted related phrases R of g_j. A bit is set if the related phrase g_k in R is in the same cluster as phrase g_j. More generally, this means that the corresponding bit in the cluster bit vector is set if there is information gain in either direction between g_j and g_k.

[0101] The cluster number then is the value of the bit string that results. This implementation has the property that related phrases that have multiple or one-way information gain appear in the same cluster.

[0102] An example of the cluster bit vectors are as follows, using the above phrases:

	Bill Clinton	President	Monica Lewinsky	purse designer	Cluster ID
Bill Clinton	1	1	1	0	14
President	1	1	0	0	12
Monica Lewinsky	1	0	1	1	11
purse designer	0	0	1	1	3

[0103] To summarize then, after this process there will be identified for each good phrase g_j, a set of related phrases R, which are sorted in order of information gain $I(g_j, g_k)$ from highest to lowest. In addition, for each good phrase g_j, there will be a cluster bit vector, the value of which is a cluster number identifying the primary cluster of which the phrase g_j is a member, and the orthogonality values (1 or 0 for each bit position) indicating which of the related phrases in R are in common clusters with g_j.

Thus in the above example, "Bill Clinton", "President", and "Monica Lewinsky" are in cluster 14 based on the values of the bits in the row for phrase "Bill Clinton".

[0104] To store this information, two basic representations are available. First, as indicated above, the information may be stored in the co-occurrence matrix 212, wherein:

[0105] entry G[row j, col. k] = (I(j,k), clusterNumber, clusterBitVector)

[0106] Alternatively, the matrix representation can be avoided, and all information stored in the good phrase list 208, wherein each row therein represents a good phrase g_j:

[0107] Phrase row_j = list [phrase g_k, (I(j,k), clusterNumber, clusterBitVector)].

[0108] This approach provides a useful organization for clusters. First, rather than a strictly--and often arbitrarily--defined hierarchy of topics and concepts, this approach recognizes that topics, as indicated by related phrases, form a complex graph of relationships, where some phrases are related to many other phrases, and some phrases have a more limited scope, and where the relationships can be mutual (each phrase predicts the other phrase) or one-directional (one phrase predicts the other, but not vice versa). The result is that clusters can be characterized "local" to each good phrase, and some clusters will then overlap by having one or more common related phrases.

[0109] For a given good phrase g_j then the ordering of the related phrases by information gain provides a taxonomy for naming the clusters of the phrase: the cluster name is the name of the related phrase in the cluster having the highest information gain.

[0110] The above process provides a very robust way of identifying significant phrases that appear in the document collection, and beneficially, the way these related phrases are used together in natural "clusters" in actual practice. As a result, this data-driven clustering of related phrases avoids the biases that are inherent in any manually directed "editorial" selection of related terms and concepts, as is common in many systems.

3. Indexing Documents with Phrases and Related Phrases

[0111] Given the good phrase list 208, including the information pertaining to related phrases and clusters, the next functional operation of the indexing system 110 is to index documents in the document collection with respect to the good phrases and clusters, and store the updated information in the index 150. FIG. 5 illustrates this process, in which there are the following functional stages for indexing a document:

[0112] 500: Post document to the posting lists of good phrases found in the document.

[0113] 502: Update instance counts and related phrase bit vector for related phases and secondary related phrases.

[0114] 504: Annotate documents with related phrase information.

[0115] 506: Reorder index entries according to posting list size.

[0116] These stages are now described in further detail.

[0117] A set of documents is traversed or crawled, as before; this may be the same or a different set of documents. For a given document d, traverse 500 the document word by word with a sequence window 302 of length n, from position i, in the manner described above.

[0118] In a given phrase window 302, identify all good phrases in the window, starting at position *i*. Each good phrase is denoted as g_i. Thus, g1 is the first good phrase, g2 would be the second good phrase, and so forth.

[0119] For each good phrase g_i (example g1 "President" and g4 "President of ATT") post the document identifier (e.g., the URL) to the posting list for the good phrase g_i in the index 150. This update identifies that the good phrase g_i appears in this specific document.

[0120] In one embodiment, the posting list for a phrase g_j takes the following logical form:

[0121] Phrase g_j: list: (document *d*, [list: related phase counts] [related phrase information])

[0122] For each phrase g_j there is a list of the documents *d* on which the phrase appears. For each document, there is a list of counts of the number of occurrences of the related phrases R of phrase g_j that also appear in document *d*.

[0123] In one embodiment, the related phrase information is a related phase bit vector. This bit vector may be characterized as a "bi-bit" vector, in that for each related phrase g_k there are two bit positions, g_k-1, g_k-2. The first bit position stores a flag indicating whether the related phrase g_k is present in the document *d* (i.e., the count for g_k in document *d* is greater than 0). The second bit position stores a flag that indicates whether a related phrase g_l of g_k is also present in document *d*. The related phrases g_l of a related phrase g_k of a phrase g_j are herein called the "secondary related phrases of g_j" The counts and bit positions correspond to the canonical order of the phrases in R (sorted in order of decreasing information gain). This sort order has the effect of making the related phrase g_k that is most highly predicted by g_j associated with the most

significant bit of the related phrase bit vector, and the related phrase g_l that is least predicted by g_j associated with the least significant bit.

[0124] It is useful to note that for a given phrase g, the length of the related phrase bit vector, and the association of the related phrases to the individual bits of the vector, will be the same with respect to all documents containing g. This implementation has the property of allowing the system to readily compare the related phrase bit vectors for any (or all) documents containing g, to see which documents have a given related phrase. This is beneficial for facilitating the search process to identify documents in response to a search query. Accordingly, a given document will appear in the posting lists of many different phrases, and in each such posting list, the related phrase vector for that document will be specific to the phrase that owns the posting list. This aspect preserves the locality of the related phrase bit vectors with respect to individual phrases and documents.

[0125] Accordingly, the next stage 502 includes traversing the secondary window 304 of the current index position in the document (as before a secondary window of +/- K terms, for example, 30 terms), for example from i-K to i+K. For each related phrase g_k of g_i that appears in the secondary window 304, the indexing system 110 increments the count of g_k with respect to document *d* in the related phrase count. If g_i appears later in the document, and the related phrase is found again within the later secondary window, again the count is incremented.

[0126] As noted, the corresponding first bit g_k-1 in the related phrase bit map is set based on the count, with the bit set to 1 if the count for g_k is >0, or set to 0 if the count equals 0.

[0127] Next, the second bit, g_k-2 is set by looking up related phrase g_k in the index 150, identifying in g_k's posting list the entry for document *d*, and then checking the secondary related phrase counts (or bits) for g_k for any its related phrases. If any of these secondary related phrases counts/bits are set, then this indicates that the secondary related phrases of g_j are also present in document *d*.

[0128] When document *d* has been completely processed in this manner, the indexing system 110 will have identified the following:

[0129] i) each good phrase g_j in document *d;*

[0130] ii) for each good phrase g_j which of its related phrases g_k are present in document *d*;

[0131] iii) for each related phrase g_k present in document d, which of its related phrases g_l (the secondary related phrases of g_j) are also present in document *d*.

a) Determining the Topics for a Document

[0132] The indexing of documents by phrases and use of the clustering information provides yet another advantage of the indexing system 110, which is the ability to determine the topics that a document is about based on the related phrase information.

[0133] Assume that for a given good phrase g_j and a given document *d*, the posting list entry is as follows:

[0134] g_j: document d: related phrase counts:= {3,4,3,0,0,2,1,1,0}

[0135] related phrase bit vector:={ 11 11 10 00 00 10 10 10 01}

[0136] where, the related phrase bit vector is shown in the bi-bit pairs.

[0137] From the related phrase bit vector, we can determine primary and secondary topics for the document *d*. A primary topic is indicated by a bit pair (1,1), and

a secondary topic is indicated by a bit pair (1,0). A related phrase bit pair of (1,1) indicates that both the related phrase g_k for the bit pair is present in document d, along the secondary related phrases g_l as well. This may be interpreted to mean that the author of the document d used several related phrases g_j, g_k, and g_l together in drafting the document. A bit pair of (1,0) indicates that both g_j and g_k are present, but no further secondary related phrases from g_k are present, and thus this is a less significant topic.

b) Document Annotation for Improved Ranking

[0138] A further aspect of the indexing system 110 is the ability to annotate 504 each document d during the indexing process with information that provides for improved ranking during subsequent searches. The annotation process 506 is as follows.

[0139] A given document d in the document collection may have some number of outlinks to other documents. Each outlink (a hyperlink) includes anchor text and the document identifier of the target document. For purposes of explanation, a current document d being processed will be referred to as URL0, and the target document of an outlink on document d will be referred to as URL1. For later use in ranking documents in search results, for every link in URL0, which points to some other URLi, the indexing system 110 creates an outlink score for the anchor phrase of that link with respect to URL0, and an inlink score for that anchor phrase with respect to URLi. That is, each link in the document collection has a pair of scores, an outlink score and an inlink score. These scores are computed as follows.

[0140] On a given document URL0, the indexing system 110 identifies each outlink to another document URL1, in which the anchor text A is a phrase in the good

phrase list 208. FIG. 7a illustrates schematically this relationship, in which anchor text "A" in document URL0 is used in a hyperlink 700.

[0141] In the posting list for phrase A, URL0 is posted as an outlink of phrase A, and URL1 is posted as an inlink of phrase A. For URL0, the related phrase bit vector is completed as described above, to identify the related phrases and secondary related phrases of A present in URL0. This related phrase bit vector is used as the outlink score for the link from URL0 to URL1 containing anchor phrase A.

[0142] Next, the inlink score is determined as follows. For each inlink to URL1 containing the anchor phrase A, the indexing system 110 scans URL1, and determines whether phrase A appears in the body of URL1. If phrase A not only points to URL1 (via a outlink on URL0), but also appears in the content of URL1 itself, this suggests that URL1 can be said to be intentionally related to the concept represented by phrase A. FIG. 7b illustrates this case, where phrase A appears in both URL0 (as anchor text) and in the body of URL1. In this case, the related phrase bit vector for phrase A for URL1 is used as the inlink score for the link from URL0 to URL1 containing phrase A.

[0143] If the anchor phrase A does not appear in the body of URL1 (as in FIG. 8a), then a different step is taken to determine the inlink score. In this case, the indexing system 110 creates a related phrase bit vector for URL1 for phrase A (as if phrase A was present in URL1) and indicating which of the related phrases of phrase A appear in URL1. This related phrase bit vector is then used as the inlink score for the link from URL0 to URL1.

[0144] For example, assume the following phrases are initially present in URL0 and URL1:

Document	Anchor Phrase	Related Phrase Bit Vector				
	Australian Shepherd	Aussie	blue merle	red merle	tricolor	agility training
URL0	1	1	0	0	0	0
URL1	1	0	1	1	1	0

[0145] (In the above, and following tables, the secondary related phrase bits are not shown). The URL0 row is the outlink score of the link from anchor text A, and the URL1 row is the inlink score of the link. Here, URL0 contains the anchor phrase "Australian Shepard" which targets URL1. Of the five related phrases of "Australian Shepard", only one, "Aussie" appears in URL0. Intuitively then, URL0 is only weakly about Australian Shepherds. URL1, by comparison, not only has the phrase "Australian Shepherd" present in the body of the document, but also has many of the related phrases present as well, "blue merle," "red merle," and "tricolor." Accordingly, because the anchor phrase "Australian Shepard" appears in both URL0 and URL1, the outlink score for URL0, and the inlink score for URL1 are the respective rows shown above.

[0146] The second case described above is where anchor phrase A does not appear in URL1. In that, the indexing system 110 scans URL1 and determines which of the related phrases "Aussie," "blue merle," "red merle," "tricolor," and "agility training" are present in URL1, and creates an related phrase bit vector accordingly, for example:

Document	Anchor Phrase	Related Phrase Bit Vector				
	Australian Shepherd	Aussie	blue merle	red merle	tricolor	agility training
URL0	1	1	0	0	0	0
URL1	0	0	1	1	1	0

Here, this shows that the URL1 does not contain the anchor phrase "Australian Shepard", but does contain the related phrases "blue merle", "red merle", and "tricolor". This approach has the benefit of entirely preventing certain types of manipulations of web pages (a class of documents) in order to skew the results of a search. Search engines that use a ranking algorithm that relies on the number of links that point to a given document in order to rank that document can be "bombed" by artificially creating a large number of pages with a given anchor text which then point to a desired page. As a result, when a search query using the anchor text is entered, the desired page is typically returned, even if in fact this page has little or nothing to do with the anchor text. Importing the related bit vector from a target document URL1 into the phrase A related phrase bit vector for document URL0 eliminates the reliance of the search system on just the relationship of phrase A in URL0 pointing to URL1 as an indicator of significance or URL1 to the anchor text phrase.

Each phrase in the index 150 is also given a phrase number, based on its frequency of occurrence in the corpus. The more common the phrase, the lower phrase number it receivesorder in the index. The indexing system 110 then sorts 506 all of the posting lists in the index 150 in declining order according to the number of documents listedphrase number of in each posting list, so that the most frequently occurring phrases are listed first. The phrase number can then be used to look up a particular phrase.

III. Search System

[0147] The search system 120 operates to receive a query and search for documents relevant to the query, and provide a list of these documents (with links to the documents) in a set of search results. FIG. 6 illustrates the main functional operations of the search system 120:

[0148] 600: Identify phrases in the query.

[0149] 602: Retrieve documents relevant to query phrases.

[0150] 604: Rank documents in search results according to phrases.

[0151] The details of each of these of these stages is as follows.

1. Identification of Phrases in the Query and Query Expansion

[0152] The first stage 600 of the search system 120 is to identify any phrases that are present in the query in order to effectively search the index. The following terminology is used in this section:

[0153] q: a query as input and receive by the search system 120.

[0154] Qp: phrases present in the query.

[0155] Qr: related phrases of Qp.

[0156] Qe: phrase extensions of Qp.

[0157] Q: the union of Qp and Qr.

[0158] A query q is received from a client 190, having up to some maximum number of characters or words.

[0159] A phrase window of size N (e.g., 5) is used by the search system 120 to traverse the terms of the query q. The phrase window starts with the first term of the query, extends N terms to the right. This window is then shifted right M-N times, where M is the number of terms in the query.

[0160] At each window position, there will be N terms (or fewer) terms in the window. These terms constitute a possible query phrase. The possible phrase is looked up in the good phrase list 208 to determine if it is a good phrase or not. If the possible phrase is present in the good phrase list 208, then a phrase number is returned for phrase; the possible phrase is now a candidate phrase.

[0161] After all possible phrases in each window have been tested to determine if they are good candidate phrases, the search system 120 will have a set of phrase numbers for the corresponding phrases in the query. These phrase numbers are then sorted (declining order).

[0162] Starting with the highest phrase number as the first candidate phrase, the search system 120 determines if there is another candidate phrase within a fixed numerical distance within the sorted list, i.e., the difference between the phrase numbers is within a threshold amount, e.g. 20,000. If so, then the phrase that is leftmost in the query is selected as a valid query phrase Qp. This query phrase and all of its sub-phrases is removed from the list of candidates, and the list is resorted and the process repeated. The result of this process is a set of valid query phrases Qp.

[0163] For example, assume the search query is "Hillary Rodham Clinton Bill on the Senate Floor". The search system 120 would identify the following candidate phrases, "Hillary Rodham Clinton Bill on," "Hillary Rodham Clinton Bill," and "Hillary Rodham Clinton". The first two are discarded, and the last one is kept as a valid query phrase. Next the search system 120 would identify "Bill on the Senate Floor", and the subsphrases "Bill on the Senate", "Bill on the", "Bill on", "Bill", and would select "Bill" as a valid query phrase Qp. Finally, the search system 120 would parse "on the senate floor" and identify "Senate Floor" as a valid query phrase.

[0164] Next, the search system 120 adjusts the valid phrases Qp for capitalization. When parsing the query, the search system 120 identifies potential capitalizations in each valid phrase. This may be done using a table of known capitalizations, such as "united states" being capitalized as "United States", or by using

a grammar based capitalization algorithm. This produces a set of properly capitalized query phrases.

[0165] The search system 120 then makes a second pass through the capitalized phrases, and selects only those phrases are leftmost and capitalized where both a phrase and its subphrase is present in the set. For example, a search on "president of the united states" will be capitalized as "President of the United States".

[0166] In the next stage, the search system 120 identifies 602 the documents that are relevant to the query phrases Q. The search system 120 then retrieves the posting lists of the query phrases Q, and intersects these lists to determine which documents appear on the all (or some number) of the posting lists for the query phrases. If a phrase Q in the query has a set of phrase extensions Qe (as further explained below), then the search system 120 first forms the union of the posting lists of the phrase extensions, prior to doing the intersection with the posting lists. The search system 120 identifies phrase extensions by looking up each query phrase Q in the incomplete phrase list 216, as described above.

[0167] The result of the intersection is a set of documents that are relevant to the query. Indexing documents by phrases and related phrases, identifying phrases Q in the query, and then expanding the query to include phrase extensions results in the selection of a set of documents that are more relevant to the query then would result in a conventional Boolean based search system in which only documents that contain the query terms are selected.

[0168] In one embodiment, the search system 120 can use an optimized mechanism to identify documents responsive to the query without having to intersect all of the posting lists of the query phrases Q. As a result of the structure of the index 150,

for each phrase g_j, the related phrases g_k are known and identified in the related phrase bit vector for g_k. Accordingly, this information can be used to shortcut the intersection process where two or more query phrases are related phrases to each other, or have common related phrases. In those cases, the related phrase bit vectors can be directly accessed, and then used next to retrieve corresponding documents. This process is more fully described as follows.

[0169] Given any two query phrases Q1 and Q2, there are three possible cases of relations:

[0170] 1) Q2 is a related phrase of Q1;

[0171] 2) Q2 is not a related phrase of Q1 and their respective related phrases Qr1 and Qr2 do not intersect (i.e., no common related phrases); and

[0172] 3) Q2 is not a related phrase of Q1, but their respective related phrases Qr1 and Qr2 do intersect.

[0173] For each pair of query phrases the search system 120 determines the appropriate case by looking up the related phrase bit vector of the query phrases Qp.

[0174] The search system 120 proceeds by retrieving the posting list for query phrase Q1, which contains the documents containing Q1, and for each of these documents, a related phrase bit vector. The related phrase bit vector for Q1 will indicated whether phrase Q2 (and each of the remaining query phrases, if any) is a related phrase of Q1 and is present in the document.

[0175] If the first case applies to Q2, the search system 120 scans the related phrase bit vector for each document d in Q1's posting list to determine if it has a bit set for Q2. If this bit is not set in for document d in Q1's posting list, then it means that Q2 does not appear in that document. As result, this document can be immediately

eliminated from further consideration. The remaining documents can then be scored. This means further that it is unnecessary for the search system 120 to process the posting lists of Q2 to see which documents it is present in as well, thereby saving compute time.

[0176] If the second case applies to Q2, then the two phrases are unrelated to each other. For example the query "cheap bolt action rifle" has two phrases "cheap" and "bolt action rifle". Neither of these phrases is related to each other, and further the related phrases of each of these do not overlap; i.e., "cheap" has related phrases "low cost," "inexpensive," "discount," "bargain basement," and "lousy,", whereas "bolt action rifle" has related phrases "gun," "22 caliber", "magazine fed," and "Armalite AR-30M", which lists thus do not intersect. In this case, the search system 120 does the regular intersection of the posting lists of Q1 and Q2 to obtain the documents for scoring.

[0177] If the third case applies, then here the two phrases Q1 and Q2 that are not related, but that do have at least one related phrase in common. For example the phrases "bolt action rifle" and "22" would both have "gun" as a related phase. In this case, the search system 120 retrieves the posting lists of both phrases Q1 and Q2 and intersects the lists to produce a list of documents that contain both phrases.

[0178] The search system 120 can then quickly score each of the resulting documents. First, the search system 120 determines a score adjustment value for each document. The score adjustment value is a mask formed from the bits in the positions corresponding to the query phrases Q1 and Q2 in the related phrase bit vector for a document. For example, assume that Q1 and Q2 correspond to the 3rd and 6th bi-bit positions in the related phrase bit vector for document d, and the bit values in 3rd position are (1,1) and the bit values in the 6th pair are (1,0), then the score adjustment

value is the bit mask "00 00 11 00 00 10". The score adjustment value is then used to mask the related phrase bit vector for the documents, and modified phrase bit vectors then are passed into the ranking function (next described) to be used in calculating a body score for the documents.

2. Ranking

a) Ranking Documents Based on Contained Phrases

[0179] The search system 120 provides a ranking stage 604 in which the documents in the search results are ranked, using the phrase information in each document's related phrase bit vector, and the cluster bit vector for the query phrases. This approach ranks documents according to the phrases that are contained in the document, or informally "body hits."

[0180] As described above, for any given phrase g_j, each document d in the g_j's posting list has an associated related phrase bit vector that identifies which related phrases g_k and which secondary related phrases g_l are present in document d. The more related phrases and secondary related phrases present in a given document, the more bits that will be set in the document's related phrase bit vector for the given phrase. The more bits that are set, the greater the numerical value of the related phrase bit vector.

[0181] Accordingly, in one embodiment, the search system 120 sorts the documents in the search results according to the value of their related phrase bit vectors. The documents containing the most related phrases to the query phrases Q will have the highest valued related phrase bit vectors, and these documents will be the highest-ranking documents in the search results.

[0182] This approach is desirable because semantically, these documents are most topically relevant to the query phrases. Note that this approach provides highly

relevant documents even if the documents do not contain a high frequency of the input query terms q, since related phrase information was used to both identify relevant documents, and then rank these documents. Documents with a low frequency of the input query terms may still have a large number of related phrases to the query terms and phrases and thus be more relevant than documents that have a high frequency of just the query terms and phrases but no related phrases.

[0183] In a second embodiment, the search system 120 scores each document in the result set according which related phrases of the query phrase Q it contains. This is done as follows:

[0184] Given each query phrase Q, there will be some number N of related phrases Qr to the query phrase, as identified during the phrase identification process. As described above, the related query phrases Qr are ordered according to their information gain from the query phrase Q. These related phrases are then assigned points, started with N points for the first related phrase Qr1 (i.e., the related phrase Qr with the highest information gain from Q), then N-1 points for the next related phrase Qr2, then N-2 points for Qr3, and so on, so that the last related phrase QrN is assigned 1 point.

[0185] Each document in the search results is then scored by determining which related phrases Qr of the query phrase Q are present, and giving the document the points assigned to each such related phrase Qr. The documents are then sorted from highest to lowest score.

[0186] As a further refinement, the search system 120 can cull certain documents from the result set. In some cases documents may be about many different topics; this is particularly the case for longer documents. In many cases, users prefer documents that

are strongly on point with respect to a single topic expressed in the query over documents that are relevant to many different topics.

[0187] To cull these latter types of documents, the search system 120 uses the cluster information in the cluster bit vectors of the query phrases, and removes any document in which there are more than a threshold number of clusters in the document. For example, the search system 120 can remove any documents that contain more than two clusters. This cluster threshold can be predetermined, or set by the user as a search parameter.

b) Ranking Documents Based on Anchor Phrases

[0188] In addition to ranking the documents in the search results based on body hits of query phrases Q, in one embodiment, the search system 120 also ranks the documents based on the appearance of query phrases Q and related query phrases Qr in anchors to other documents. In one embodiment, the search system 120 calculates a score for each document that is a function (e.g., linear combination) of two scores, a body hit score and an anchor hit score.

[0189] For example, the document score for a given document can be calculated as follows:

[0190] Score = .30*(body hit score)+.70*(anchor hit score).

[0191] The weights of .30 and .70 can be adjusted as desired. The body hit score for a document is the numerical value of the highest valued related phrase bit vector for the document, given the query phrases Qp, in the manner described above. Alternatively, this value can directly obtained by the search system 120 by looking up each query phrase Q in the index 150, accessing the document from the posting list of the query phrase Q, and then accessing the related phrase bit vector.

[0192] The anchor hit score of a document *d* a function of the related phrase bit vectors of the query phrases Q, where Q is an anchor term in a document that references document *d*. When the indexing system 110 indexes the documents in the document collection, it maintains for each phrase a list of the documents in which the phrase is anchor text in an outlink, and also for each document a list of the inlinks (and the associated anchor text) from other documents. The inlinks for a document are references (e.g. hyperlinks) from other documents (referencing documents) to a given document.

[0193] To determine the anchor hit score for a given document *d* then, the search system 120 iterates over the set of referencing documents R (i=1 to number of referencing documents) listed in index by their anchor phrases Q, and sums the following product:

[0194] R_i.Q.Related phrase bit vector*D.Q.Related phrase bit vector.

[0195] The product value here is a score of how topical anchor phrase Q is to document D. This score is here called the "inbound score component." This product effectively weights the current document D's related bit vector by the related bit vectors of anchor phrases in the referencing document R. If the referencing documents R themselves are related to the query phrase Q (and thus, have a higher valued related phrase bit vector), then this increases the significance of the current document D score. The body hit score and the anchor hit score are then combined to create the document score, as described above.

[0196] Next, for each of the referencing documents R, the related phrase bit vector for each anchor phrase Q is obtained. This is a measure of how topical the anchor phrase Q is to the document R. This value is here called the outbound score component.

[0197] From the index 150 then, all of the (referencing document, referenced document) pairs are extracted for the anchor phrases Q. These pairs are then sorted by their associated (outbound score component, inbound score component) values. Depending on the implementation, either of these components can be the primary sort key, and the other can be the secondary sort key. The sorted results are then presented to the user. Sorting the documents on the outbound score component makes documents that have many related phrases to the query as anchor hits, rank most highly, thus representing these documents as "expert" documents. Sorting on the inbound document score makes documents that frequently referenced by the anchor terms the most high ranked.

IV. Top Phrases and Phrase Information Refinement

[0198] The phrase information refinement system 130 uses the per-document phrase information generated by the indexing system 110 to determine additional phrase information for individual websites (or other limited document collections), and uses any modifications made by users to this additional information to refine the existing generated phrase information stored in index 150. FIG. 8 illustrates the main functional operations of the phrase information refinement system 130:

[0199] 800: Determine the top phrases associated with a given website

[0200] 810: Receive additional related phrases for top phrases

[0201] 820: Update related phrases of top phrase with additional related phrases

[0202] 830: Update additional related phrases with information from existing related phrases.

1. Determining Top Phrases

[0203] In addition to determining the documents in which particular phrases and related phrases occur , as already accomplished by the indexing system 110, the phrase information refinement system 130 is configured to determine the a set of representative or significant phrases for a particular web site or other limited document collection; these representative phrases can be generally referred to as "top phrases." The "top phrases" for a website are useful indicators of the queries for which the website is likely to be relevant, and thus provide a mechanism for improved search efficiency.

[0204] For a given website, the phrase information refinement system 130 processes 800 each document within the website to determine the top phrases per page, and then aggregates these per-page top phrases to determine the top phrases of the document collection as a whole.

(a) Per-document processing

[0205] For each document in the website, the phrase information refinement system 130 determines the phrases that appear in the document, from index 150. For each identified phrase, an importance score is calculated based on the related phrases. In one embodiment, the importance score for a phrase is a function of the summed frequency of occurrence of each of the related phrases in the document. This is readily accomplished by examining the posting list of the document, created earlier by the indexing system 110, since lists of related phrases and the frequencies of each related phrase in the document are stored within the posting list for a given phrase and document. This determination means that phrases with the most numerous related phrases will be considered to be the most representative of the given document.

(b) Determining top phrases for a website

[0206] With the top phrases for each document in the website determined, the phrase information refinement system 130 now uses this per-document information in order to determine the top phrases of the website as a whole. In one embodiment, the scores of each top phrase are summed across the documents in the website, and a number N (e.g., 10) of the phrases with the highest aggregate scores are chosen to be the top phrases for the website. In another embodiment, the scores of top phrases for a document are weighted according to their positions in the document collection. For example, in a document collection consisting of website pages, pages with shorter paths to the root of the site are given a higher weighting than pages with longer paths, on the assumption that pages closer to the root are more important than pages nested deep in the page hierarchy. The top phrases for the website are then stored in a data structure indexed by document identifier for the home page for the website.

[0207] The top phrases for the website can be recomputed on periodic basis, or on demand from the website administrator. In one embodiment, on each update the scores for a previous set of top phrases can be decayed and combined with the scores for the current set of top phrases, then the final scores determined, and sorted to identify the new top phrases. For example, the final score can be a weighted combination of 75% of the current score and 25% of a previous score. This (or other linear or non-linear) decay function enables the site to gradually change its most important phrases.

2. Receive replacement top phrases for current top phrases

[0208] The phrase information refinement system 130 also provides an interface that allows the administrators of document collections, such as webmasters, to view the top phrases and to manually change them to phrases deemed more representative of the site content. Allowing administrators to make such changes confers the dual benefits of

updating the top phrase lists with more representative phrases so that the documents in the collection will be deemed relevant to a broader range of queries, and of providing additional, reliable semantic information, as discussed below.

[0209] FIG. 9 illustrates schematically a simplified web-based user interface designed for this purpose. Webmasters or other authorized administrators first enter the appropriate identifying information, such as a username and password created during an earlier registration process and identifying them as having authority for the web site. Upon authentication of this identifying information, the phrase information refinement system 130 then displays a page such as the user interface of FIG. 9. The top phrases for the site are presented in text fields 902. The administrator can provide a different replacement phrase for any of the top phrases, and submit these replacement phrases to the system 130, with buttons 904. For example, administrators could specify that the top phrase "working dog" 906 be replaced with a more representative top phrase, such as "dog sports."

3. Updating Existing Phrase Information

[0210] Changes made by administrators represent particularly reliable knowledge about the relationships of phrases, since they are manually entered by an administrator who has authority for the document collection, and who is therefore presumably highly knowledgeable about what concepts the document collection represents. Thus, it is very valuable to capture this additional knowledge, using it to supplement the existing phrase information automatically determined by the indexing system 110 and creating a richer and more representative understanding of phrase relationships.

[0211] Initially, the phrase information refinement system 130 updates the phrase information, noting using the change of a current top phrase TP_{old} to a new, administrator-specified replacement top phrase TP_{new} as the basis of the update. Responsive to the top phrase change, a series of actions is performed, the order of which need not be performed in the particular order set forth below. Rather, the order of the actions may vary greatly in different embodiments, while still accomplishing the same result. The effect of the updating step 820 is to treat each replacement top phrase "as if" it was already present in the website. In general, this is done by adding the website to the posting list of the replacement phrase, and then updating the related phrase data for the replacement top phrase with related phrase data from the old top phrase, and other top phrases. This process is now described in more detail.

[0212] First, the root document for the website, such as the base URL of the web site, is added to the posting list for the replacement top phrase TP_{new}. This in effect associates TP_{new} with the site, treating it as if it appeared on home page of the site. This is reasonable since top phrases represent the entire document collection, rather than any particular document thereof, and thus the home page serves as a proxy for a location on the site for an occurrence of the replacement top phrase.

[0213] Another action is to add the current top phrase TP_{old} to the related phrase list of the replacement related phrase TP_{new}, and to likewise add TP_{new} to the related phrase list of TPold. This action is appropriate since the administrator has expressly indicated that the phrases are related by providing the new phrase as a replacement for the old one. This feature thus allows the system to capture the semantic relationship between the two phrases. This is done by accessing the posting list for each of the phrases TP_{old} and TP_{new}, further accessing the entry for the root document of the

document collection, such as the base URL for a web site, and then updating this entry to reflect the presence of the other phrase as a related phrase.

[0214] A further action is to determine which related phrases TP_{old} and TP_{new} have in common. Since the bits of the related phrase bit vectors of one phrase do not correspond to those of another phrase, the intersection of the related phrases cannot be determined simply by intersecting the related phrase bit vectors of two phrases. Rather, the set of actual related phrases corresponding to the bit vector bits is determined for each of TP_{old} and TP_{new}, and then the two sets are intersected, the result being the phrases that are related to both TP_{old} and TP_{new}. In one embodiment, intersecting (i.e., common) related phrases have their counts in the posting list for TP_{new} set to the counts of TP_{old}, which serves to give TP_{new} a copy of the counts for TP_{old} for their common related phrases.

[0215] For example, if the related phrases of TP_{old} are "blue merle," "red merle," and "Aussie," and the related phrases of TP_{new} are "agility training," "red merle," and "working dog," then the related phrase "red merle" is in the intersection. Thus, in the posting list for TP_{new}, the entry for the root document of the collection is accessed and the count for the related phrase "red merle" is incremented.

[0216] It is expected that some webmasters and administrators will attempt to provide a replacement phrase for a top phrase to which it is not actually semantically related; this may be done either accidentally or intentionally, for example in order to attack search results to the page. This problem can be avoided by ensuring that a replacement phrase TP_{new} has a minimal degree of semantic relationship to the TP_{old} which it is to replace. In one embodiment then, TP_{new} cannot be substituted for TP_{old} unless there is some degree of relatedness of the two phrases, e.g. at least one phrase in

common in their respective primary related phrases or their secondary related phrases. Further, in this embodiment, the phrase information refinement system 130 may additionally penalize an attempt to substitute an unrelated phrase by decrementing the counts of the related phrases of TP_{new} with respect to the website. A "decrement penalty" serves to deter an administrator from entering popular but spurious top phrases in order to attract users to the site.

[0217] Still another action is to increment the counts in the related phrase list for TP_{new} for related phrases that are also top phrases of the website. This incrementing reflects the fact that the top phrases are either already present somewhere in the document collection (in the case of automatically determined top phrases) or are at least considered to be effectively, if not actually, present (in the case of manually specified top phrases). For example, assume the top phrases in a website for cooking recipes are "baked chicken," "chicken salad," "vegetable stew" and "roast beef", and further assume that new top phrase "chicken dishes" is being used to replace for old top phrase "baked chicken." Assume as well that the related phrases of "baked chicken" are "roast chicken," "broiled chicken" and "chicken salad". Since "chicken salad" is both an existing top phrase in the website and is a related phrase of the replacement phrase "chicken dishes", the entry for "chicken salad" in the related phrase list of the phrase "chicken dishes" is incremented.

[0218] The effect of these various updating actions is to update the data structures with information as if the administrator-specified replacement phrase TP_{new} were itself present in the website and related to other phrases as indicated by its posting list related phrase entries. Even though TP_{new} may not actually be present, the fact that an administrator stated it to be a top phrase of the document collection means that such

"simulated" relationship data has a strong semantic foundation and is a valuable addition to the phrase data tracked by the system 100.

[0219] With the updates to the top phrases using replacement phrases, during the search process described above, the website will be returned in response to queries that correspond to replacement phrases (and their related phrases).

[0220] 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 performed by a single component.

[0221] Some portions of 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.

[0222] Unless specifically stated otherwise as apparent from the above discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes 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.

[0223] 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 networ*k* operating systems.

[0224] 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 is not limited to, any type of dis*k* including floppy disks, optical disks, CD-ROMs, magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, application specific integrated circuits (ASICs), or any type of media suitable for

storing electronic instructions, and each coupled to a computer system bus. Furthermore, the computers referred to in the specification may include a single processor or may be architectures employing multiple processor designs for increased computing capability.

[0225] 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, 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.

[0226] The present invention is well suited to a wide variety of computer network systems over numerous topologies. Within this field, the configuration and management of large networks comprise storage devices and computers that are communicatively coupled to dissimilar computers and storage devices over a network, such as the Internet.

[0227] 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, which is set forth in the following claims.

I claim:

1. A method for updating phrases associated with a limited document collection, comprising:

determining a list of top phrases for the limited document collection, at least in part based on presence of related phrases of the top phrases;

receiving a replacement phrase for at least one of the top phrases; and

updating related phrase data for the replacement phrase from the related phrase data of the top phrase that is being replaced.

2. A method of determining top phrases of a limited document collection, comprising:

determining top phrases for each of a plurality of documents in the limited document collection, at least in part based on presence of related phrases of the top phrases in each document, each top phrase being associated with a score;

for each top phrase of a document, determining an aggregate score for the top phrase corresponding to the top phrase's scores for documents in which it appears in the limited document collection;

selecting a set of top phrases with the highest aggregate scores.

3. The method of claim 2, wherein determining the top phrases of a document comprises:

identifying each phrase of the document; and

for each identified phrase, determining an importance value based on occurrences of related phrases of the identified phrase that are also in the document.

4. The method of claim 3, wherein determining the related phrases of an identified phrase comprises examining a posting list for the identified phrase.

5. The method of claim 3, wherein determining the importance value of an identified phrase is based on a frequency of the related phrases in the document.

6. The method of claim 2, wherein determining an aggregate score for the top phrase comprises summing the scores of the top phrase, for each document for which it is a top phrase.

7. The method of claim 6, further comprising weighting each summed score based on a location within the document collection of the document to which the summed score corresponds.

8. The method of claim 2, further comprising modifying the aggregate score by combining it with a previously determined aggregate score.

9. The method of claim 8, wherein the combination of the aggregate score and the previously determined aggregate score is accomplished by a linear combination of the two scores.

10. A method of updating existing phrase information, responsive to a user requesting a change of a current top phrase for a limited document collection to a replacement top phrase, the method comprising:

associating the replacement top phrase with a root document of the document collection;

associating the current top phrase and replacement top phrase with each other;

adding to phrase information for the replacement top phrase, phrase information for the current top phrase; and

adding to related phrase information of the replacement top phrase, related phrase information of the current top phrase.

11. The method of claim 10, wherein associating the replacement top phrase with the root document of the document collection comprises adding the root document of the document collection to a posting list for the replacement top phrase.

12. The method of claim 10, wherein associating the current top phrase and replacement top phrase with each other comprises:

adding the current top phrase to the related phrase list of the replacement top phrase; and

adding the replacement top phrase to the related phrase list of the current top phrase.

13. The method of claim 10, wherein adding to phrase information for the replacement top phrase, phrase information for the current top phrase, comprises:

determining a count corresponding to a related phrase of the current top phrase that is also a top phrase of the limited document collection; and

creating for the replacement top phrase a corresponding copy of the count;

determining each related phrase of the replacement top phrase that is also a top phrase of the limited document collection; and

updating for the replacement top phrase, a count of the determined related phrases based on a number of occurrences of the related phrases.

14. The method of claim 10, wherein adding to related phrase information for the replacement top phrase, related phrase information of the current top phrase, comprises incrementing counts of related phrases of the replacement top phrase that are also related phrases of the current top phrase.

15. The method of claim 10, further comprising denying the request to change the current top phrase to the existing top phrase, responsive to the current and replacement top phrases having no related phrases in common.

16. The method of claim 10, further comprising decrementing the counts of the related phrases of the replacement top phrase, responsive to the current and replacement top phrases having no related phrases in common.

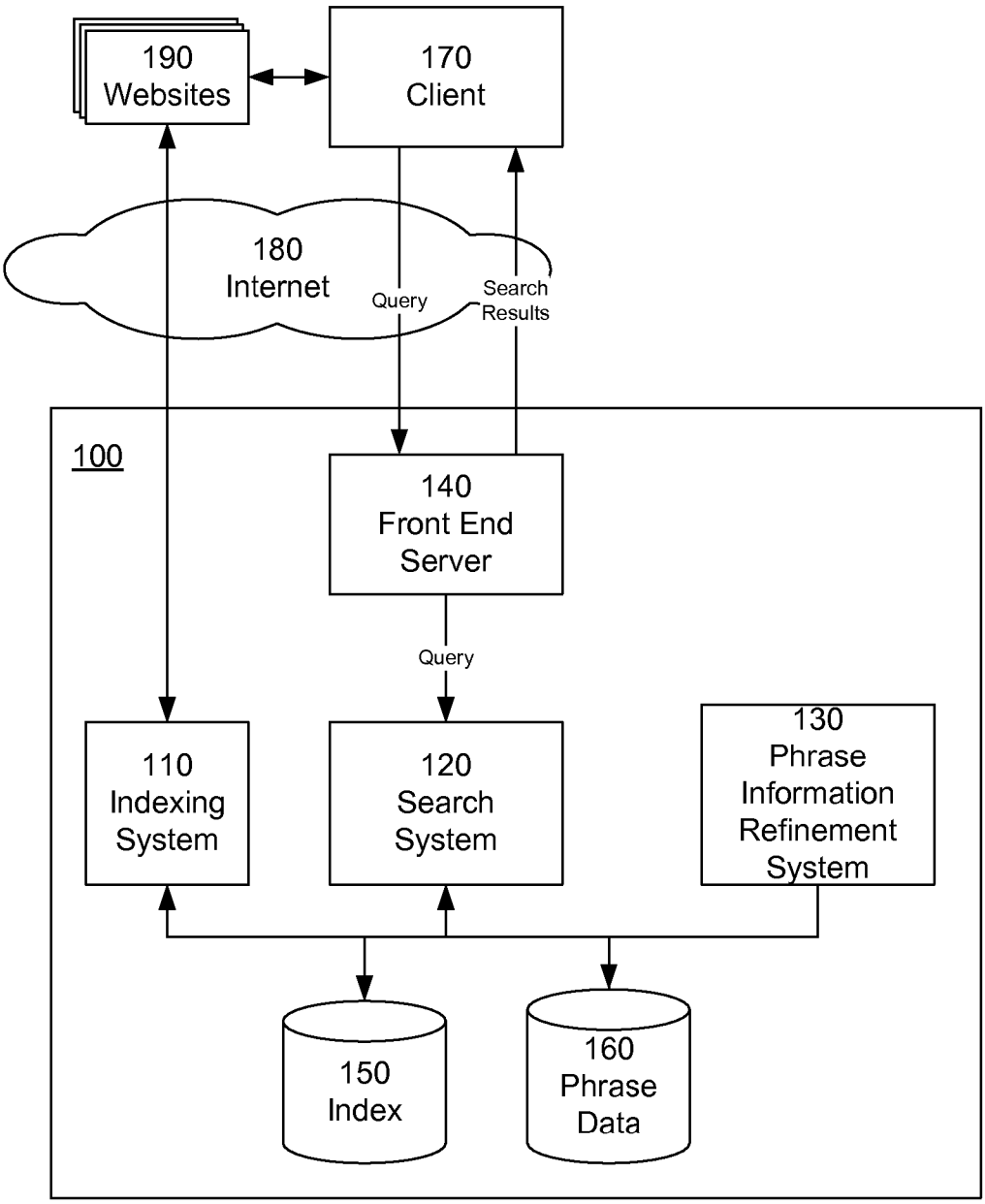
190
Websites
170
Client
180
Internet
Query
Search
Results
100
140
Front End
Server
Query
110
Indexing
System
120
Search
System
130
Phrase
Information
Refinement
System
150
Index
160
Phrase
Data

FIG. 1

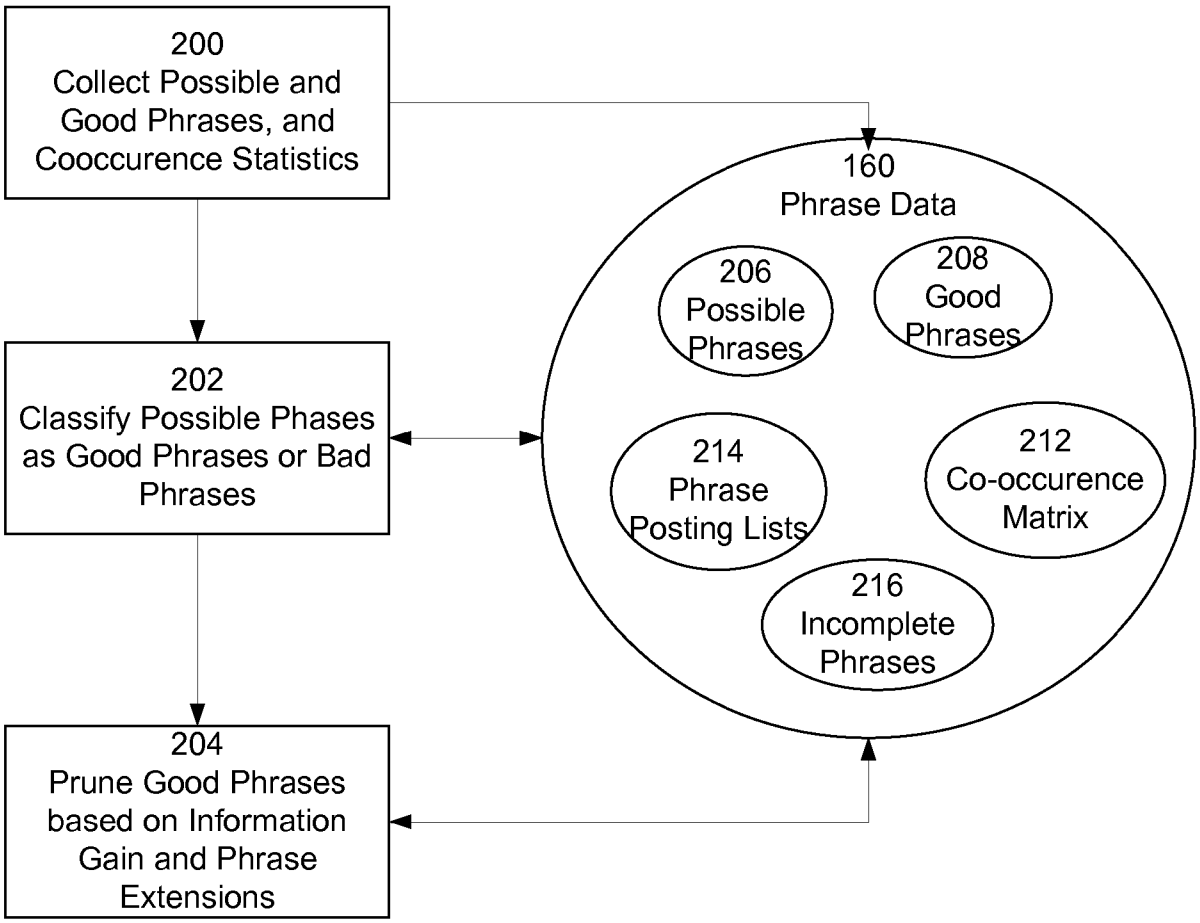
200
Collect Possible and Good Phrases, and Cooccurence Statistics
202
Classify Possible Phases as Good Phrases or Bad Phrases
204
Prune Good Phrases based on Information Gain and Phrase Extensions
160
Phrase Data
206
Possible Phrases
208
Good Phrases
214
Phrase Posting Lists
212
Co-occurence Matrix
216
Incomplete Phrases

FIG. 2

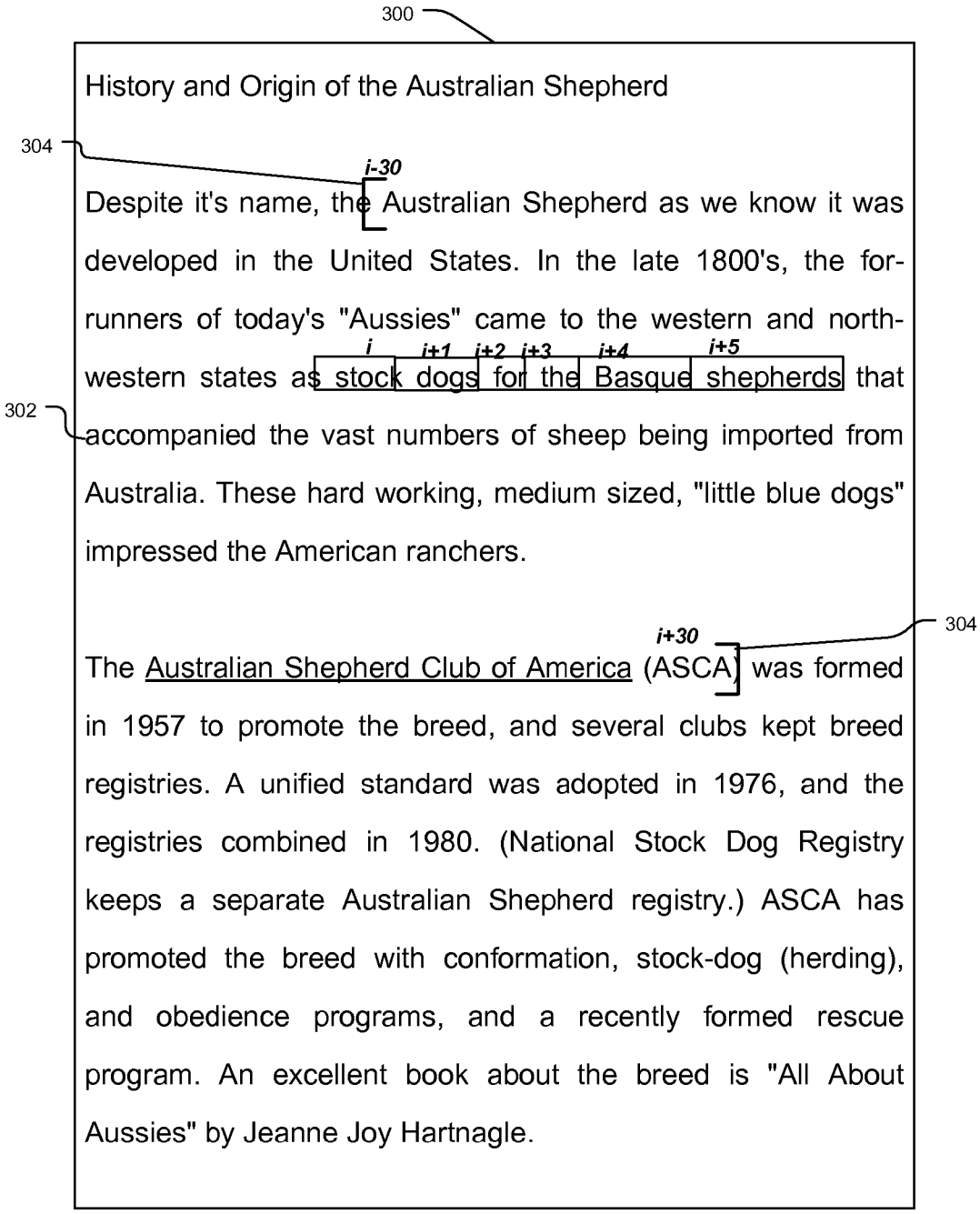
300
History and Origin of the Australian Shepherd
304
i-30
Despite it's name, the Australian Shepherd as we know it was developed in the United States. In the late 1800's, the forrunners of today's "Aussies" came to the western and northwestern states as stock dogs for the Basque shepherds that accompanied the vast numbers of sheep being imported from Australia. These hard working, medium sized, "little blue dogs" impressed the American ranchers.
i
i+1
i+2
i+3
i+4
i+5
302
i+30
304
The Australian Shepherd Club of America (ASCA) was formed in 1957 to promote the breed, and several clubs kept breed registries. A unified standard was adopted in 1976, and the registries combined in 1980. (National Stock Dog Registry keeps a separate Australian Shepherd registry.) ASCA has promoted the breed with conformation, stock-dog (herding), and obedience programs, and a recently formed rescue program. An excellent book about the breed is "All About Aussies" by Jeanne Joy Hartnagle.

FIG. 3

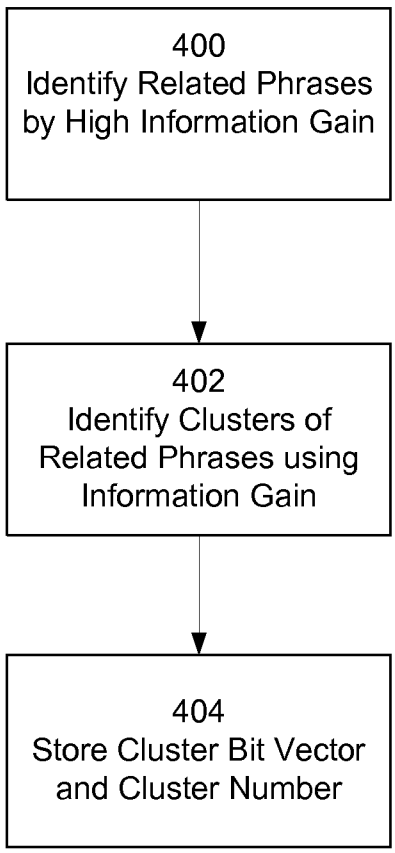
400
Identify Related Phrases by High Information Gain
402
Identify Clusters of Related Phrases using Information Gain
404
Store Cluster Bit Vector and Cluster Number

FIG. 4

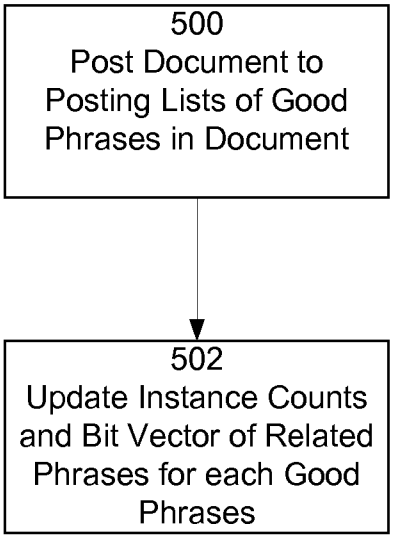
500
Post Document to Posting Lists of Good Phrases in Document
502
Update Instance Counts and Bit Vector of Related Phrases for each Good Phrases

FIG. 5

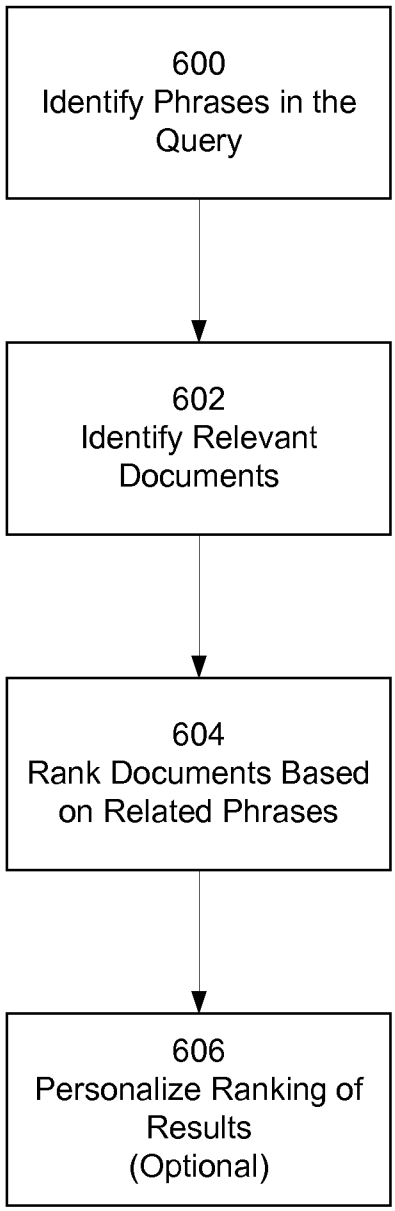
600
Identify Phrases in the Query
602
Identify Relevant Documents
604
Rank Documents Based on Related Phrases
606
Personalize Ranking of Results (Optional)

FIG. 6

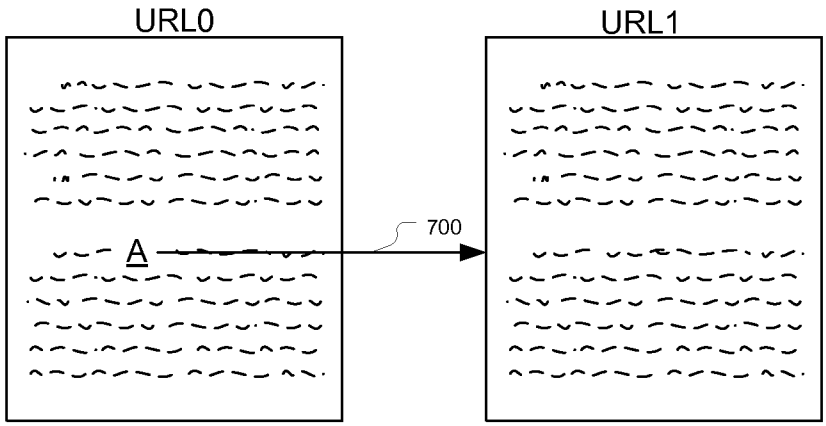

FIG. 7a

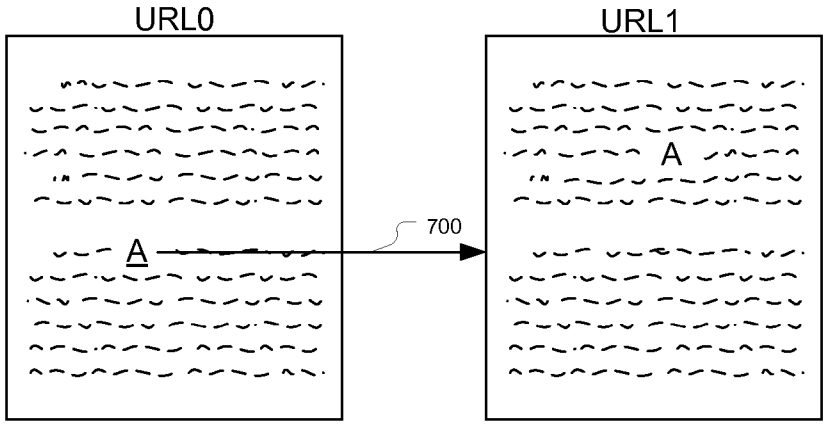

FIG. 7b

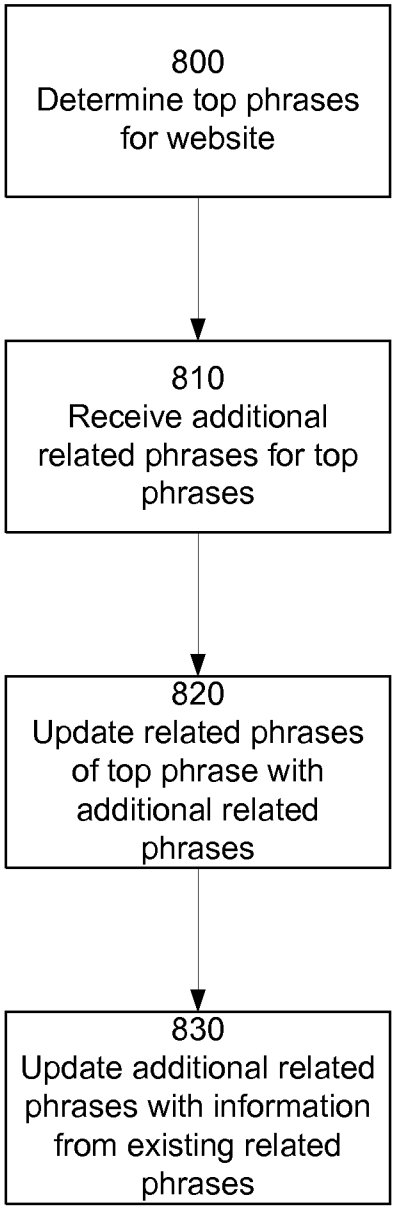
800
Determine top phrases for website
810
Receive additional related phrases for top phrases
820
Update related phrases of top phrase with additional related phrases
830
Update additional related phrases with information from existing related phrases

FIG. 8


Top phrases for site
“www.australianshepherds.com”
Agility training
Edit
Blue merle
Edit
Red merle
Edit
Aussie
Edit
Working dog
Edit
902
904
906

FIG. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 08/75492

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 7/00 (2008.04)
USPC - 707/100
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
USPC: 707/100

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 707/3, 5 (text search -- see terms below)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PubWEST(USPT, PGPB, EPAB, JPAB); DialogPRO; Google
Search Terms: System, information, retrieval, retrieve, index, organize, describe, analyze, store, identify, predict, associate, count, increment, decrement, copy, Document, collection, submitted, user, plurality, Root Document, User Request, Phrase, presence

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6,741,981 B2 (McGREEVY) 25 May 2004 (25.05.2004), FIG. 9; col 19, ln 11-19, 27-40.	1
---		---
Y		2-16
Y	US 6,823,333 B2 (McGREEVY) 23 November 2004 (23.11.2004), FIG. 16, 17, 17A, 20C and 22; col 43, ln 61-62; col 44, ln 33-40, 47-52; col 45, ln 40-42; col 46, ln 50-62; col 56, ln 16-25; col 59, ln 19-22, 25-30	2-9
Y	US 6,772,150 B1 (WHITMAN) 03 August 2004 (03.08.2004), FIG. 1, 2 and 5; col 5, ln 11-17; col 6, ln 16-25; col 7, ln 59-65; col 10, ln 40-49, 60-65; col 11, ln 4-8	10-16

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
04 November 2008 (04.11.2008)	18 NOV 2008

Name and mailing address of the ISA/US	Authorized officer:
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774