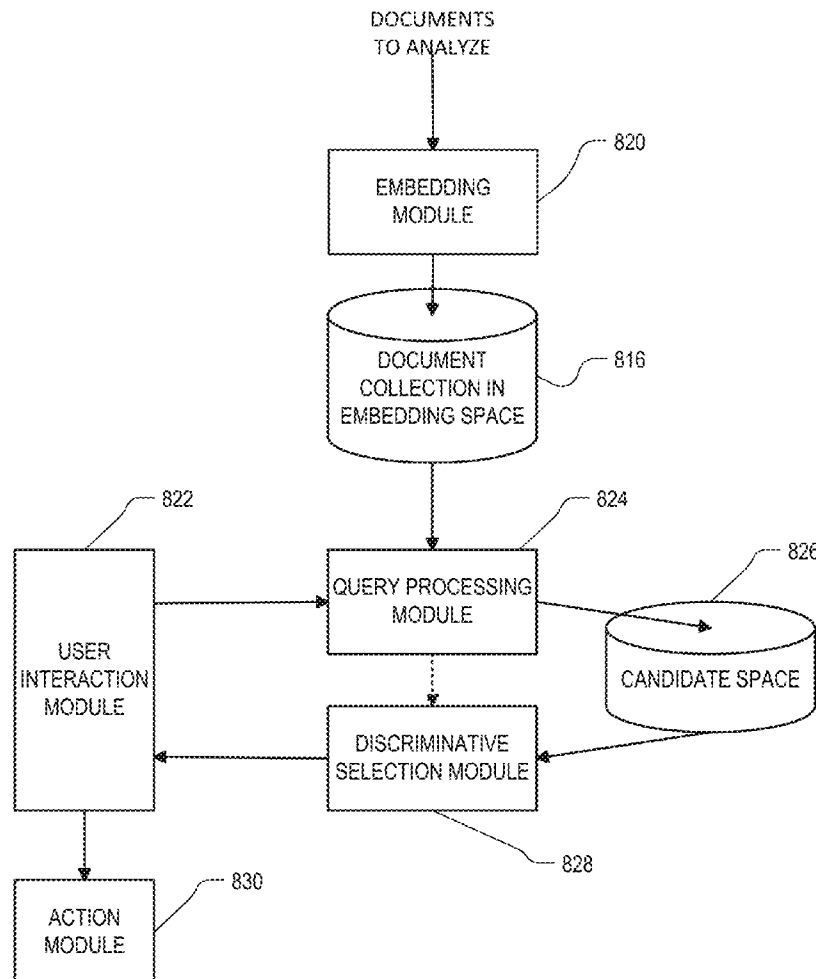




US 20150331908A1

(19) **United States**(12) **Patent Application Publication**
Duffy(10) **Pub. No.: US 2015/0331908 A1**(43) **Pub. Date: Nov. 19, 2015**(54) **VISUAL INTERACTIVE SEARCH**(71) Applicant: **Genetic Finance (Barbados) Limited,**
Belleville (BB)(72) Inventor: **Nigel Duffy,** San Francisco, CA (US)(73) Assignee: **Genetic Finance (Barbados) Limited,**
Belleville (BB)(21) Appl. No.: **14/494,364**(22) Filed: **Sep. 23, 2014****Related U.S. Application Data**(60) Provisional application No. 61/994,048, filed on May
15, 2014.**Publication Classification**(51) **Int. Cl.**
G06F 17/30 (2006.01)(52) **U.S. Cl.**CPC **G06F 17/30477** (2013.01)(57) **ABSTRACT**

Roughly described, a system and method for user identification of a desired document, in which a database is provided which identifies a collection of documents in an embedding space, the database identifying a distance between each pair of the documents in the embedding space corresponding to a predetermined measure of dissimilarity between the pair of documents. In dependence upon a user query, the system constrains the embedding space geometrically to develop a first candidate space, and identifies toward the user a first set of $N1 > 1$ candidate documents from the first candidate space, the first set of candidate documents being more discriminative than the average discriminativeness of set size $N1$ documents in the first candidate space. Preferably the placement of the documents as presented to the user is indicative of the placement of the documents in the embedding space, either in distance or in collinearity or both.



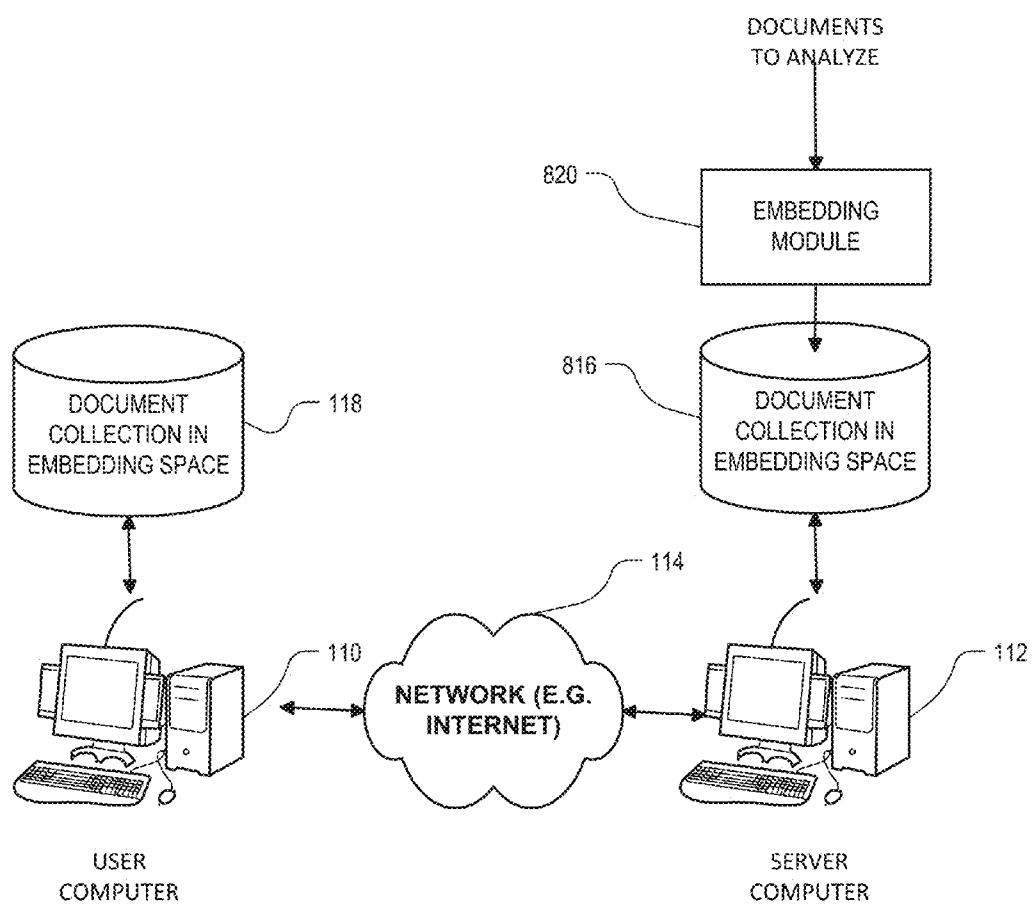


FIG. 1

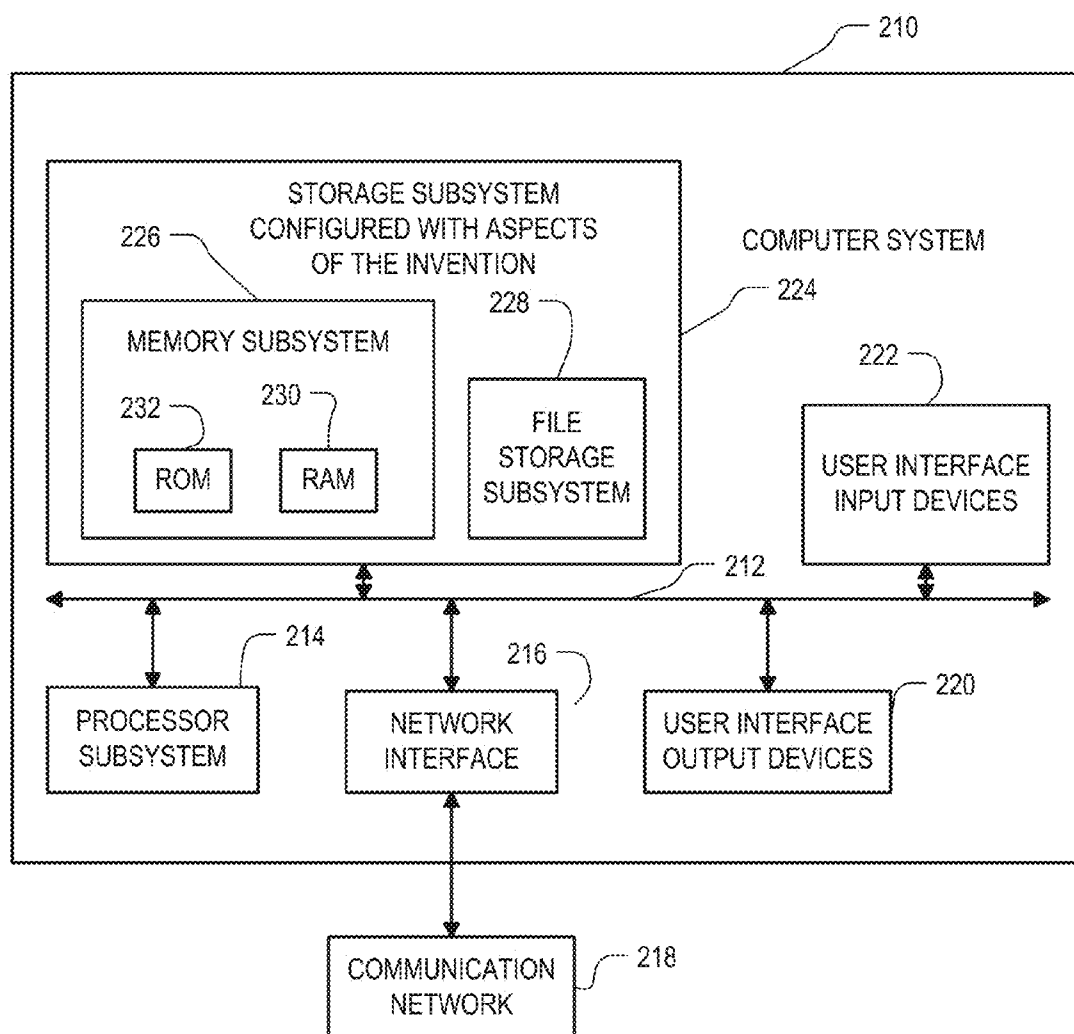


FIG. 2



FIG. 3

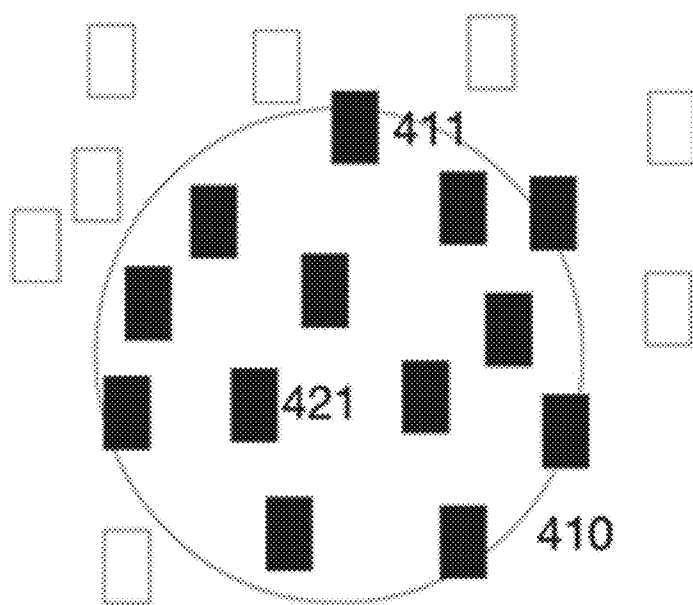


FIG. 4



FIG. 5

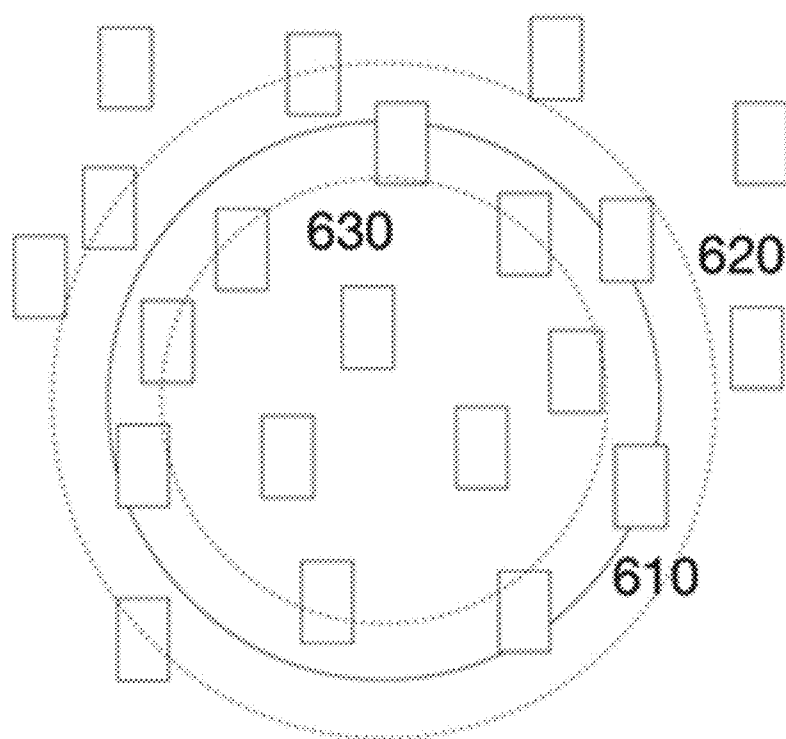


FIG. 6

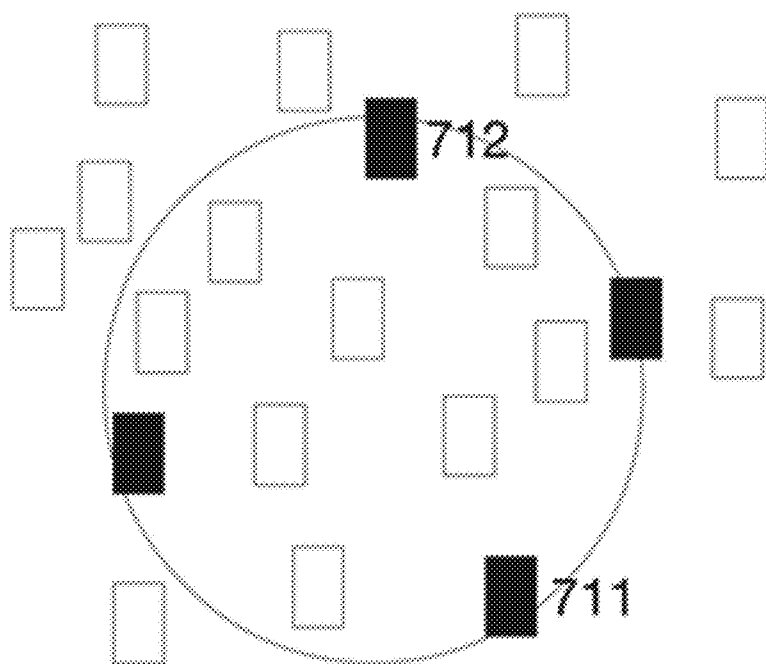


FIG. 7

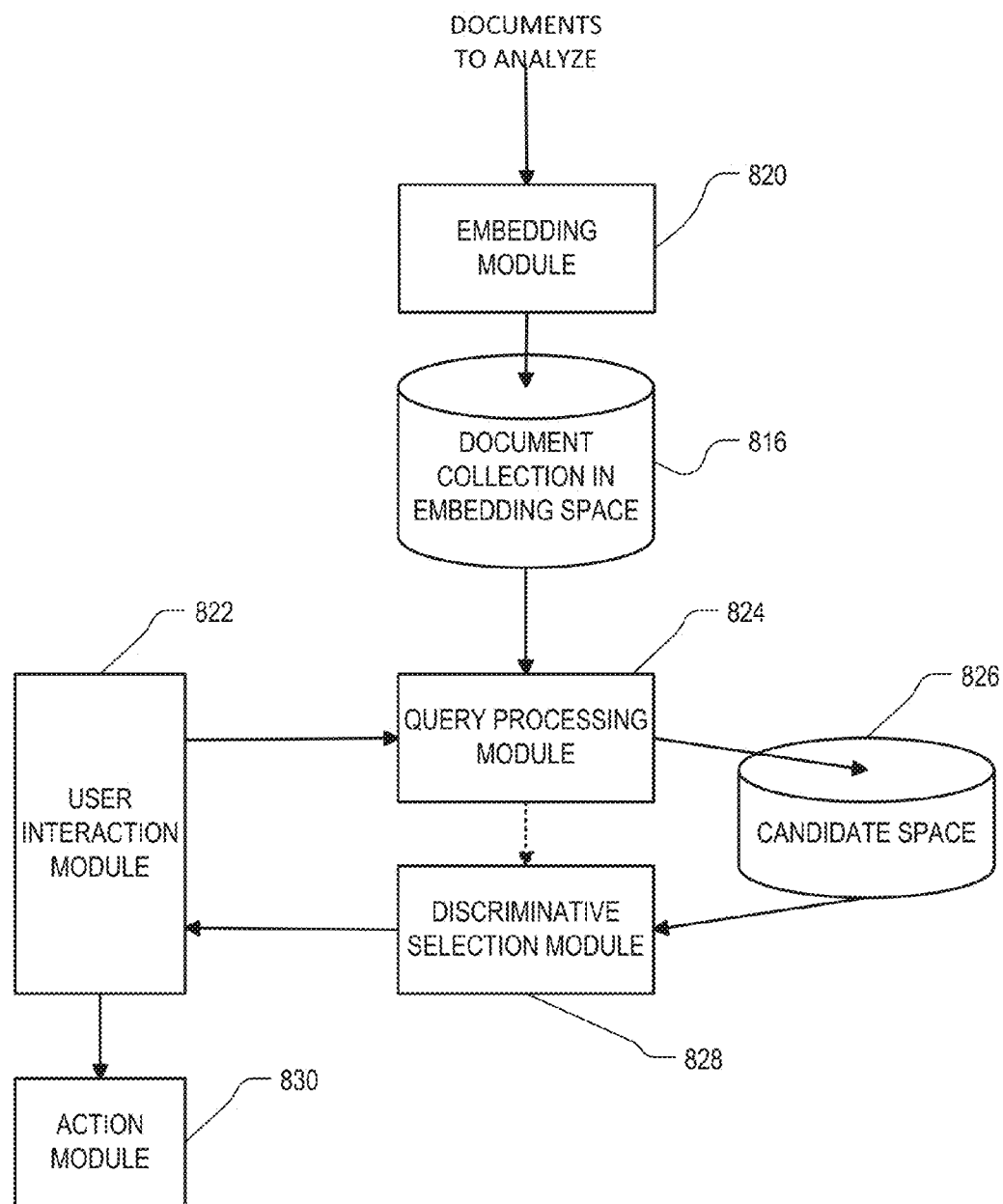


FIG. 8

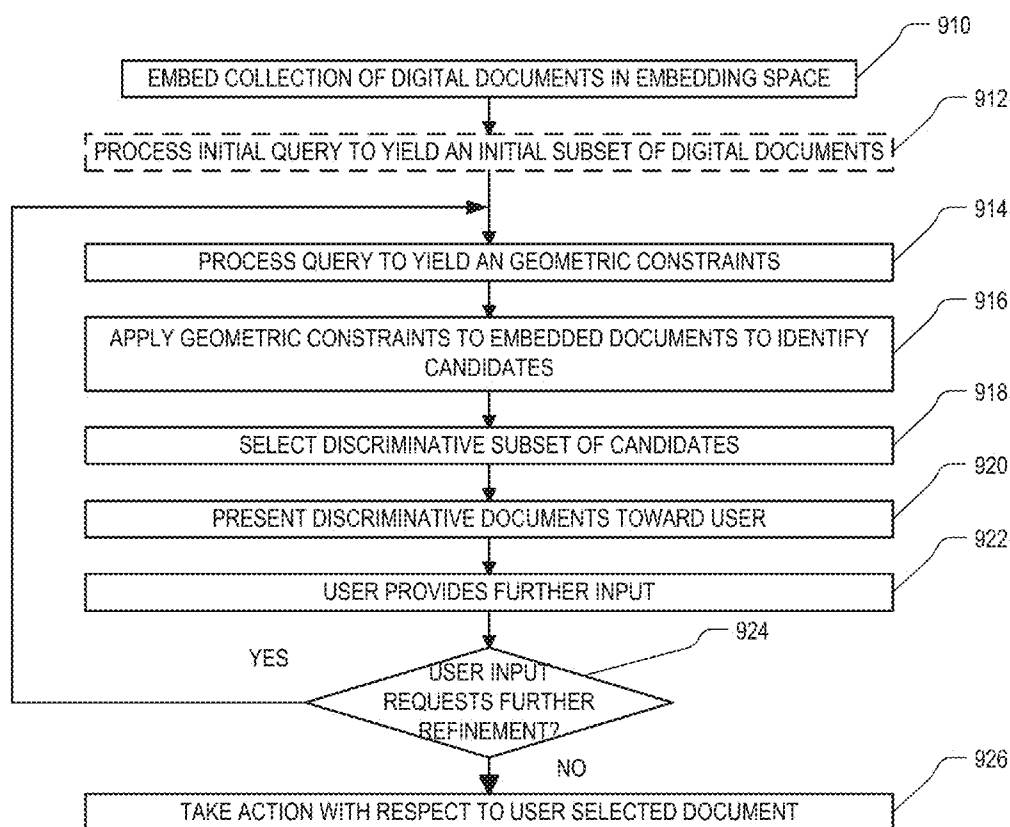
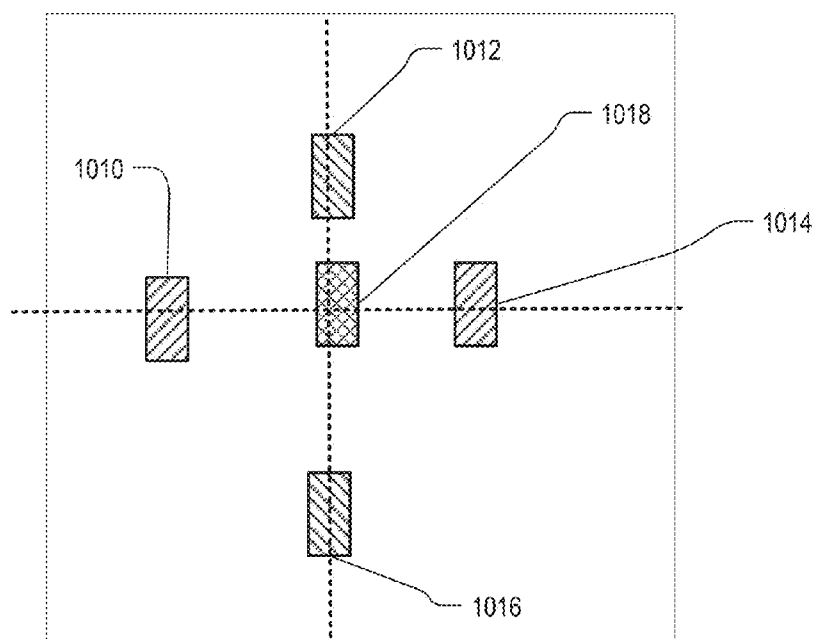
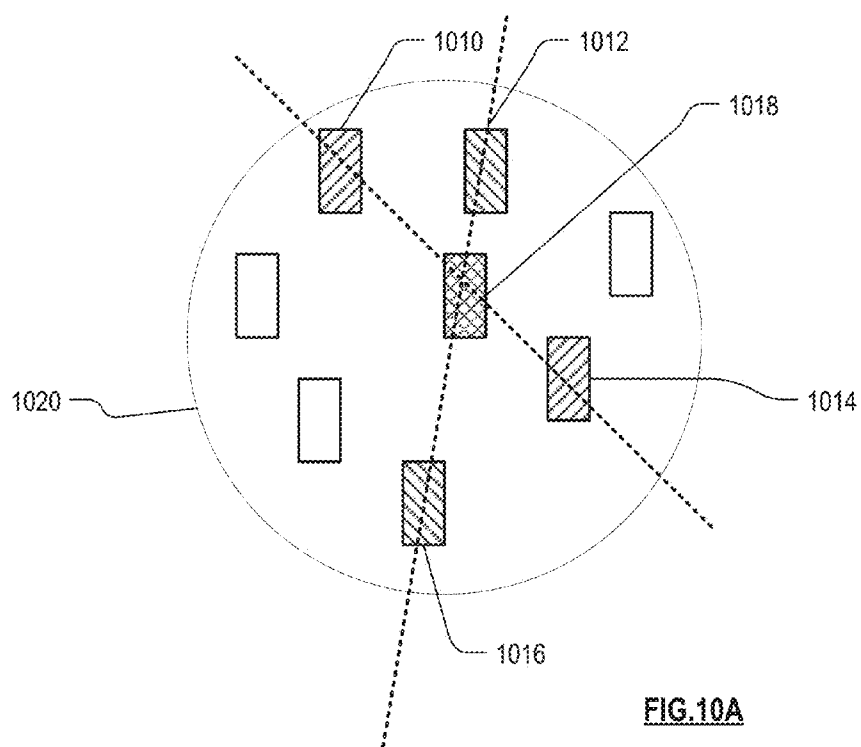


FIG. 9



VISUAL INTERACTIVE SEARCH

CROSS-REFERENCE TO OTHER APPLICATIONS

[0001] Applicants hereby claim the benefit under 35 U.S.C. 119(e) of U.S. provisional application No. 61/994,048, filed 15 May 2014. The provisional application is hereby incorporated by reference herein for its teachings.

BACKGROUND

[0002] The invention relates generally to a tool for searching for digital documents in an interactive and visual way. Examples of digital documents include: photographs, product descriptions, or webpages. For example this tool may be used on a mobile device to search for furniture available for sale via an online retailer.

[0003] Current computer search technologies allow users to perform queries and respond to those queries with an ordered list of results. The queries may be in the form of a structured query language, natural language text, speech, or a reference image. However, the results returned often do not satisfy the user's search goal. The user then proceeds to refine or modify the query in an attempt to better achieve desired goals.

SUMMARY

[0004] The system described herein addresses this challenge in at least two ways. First, it presents a set of search results that are diverse along multiple axes. Second, it allows the user to interactively "navigate" within those results by selecting results that incrementally address desired search goals and using that user input to iteratively refine the set of results presented to the user.

[0005] Roughly described, a user performs an initial search using a query, which maybe textual, verbal, or visual, e.g., via a prototype image. The system returns a set of results in response to the query. The result set is selected to be diverse, that is, each of the results differs from the others along different axes relevant to the search query. The results are presented in a 2 dimensional arrangement (such as a grid). One result, the primary result, may be displayed more prominently. The other results are displayed in a layout relative to the primary result such that results that are most similar to each other are laid out near to each other and results that are most dissimilar from each other are laid out far from each other. The user may select one or more results to indicate a request to refine the search to identify items more similar to those results. The system then returns and displays a set of results more similar to the selected results and less similar to the non-selected results. In this way the user can interactively refine a search specification in order to more precisely encode intended search goals.

[0006] The above summary of the invention is provided in order to provide a basic understanding of some aspects of the invention. This summary is not intended to identify key or critical elements of the invention or to delineate the scope of the invention. Its sole purpose is to present some concepts of the invention in a simplified form as a prelude to the more detailed description that is presented later. Particular aspects of the invention are described in the claims, specification and drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The invention will be described with respect to specific embodiments thereof, and reference will be made to the drawings, in which:

[0008] FIG. 1 illustrates an example environment in which aspects of the invention may be implemented.

[0009] FIG. 2 is a simplified block diagram of a computer system that can be used to implement aspects of the present invention.

[0010] FIGS. 3-7 and 10A illustrate documents embedded in 2-dimensional embedding space.

[0011] FIG. 8 is a block diagram of various components of an embodiment of a system according to the invention.

[0012] FIG. 9 is a flow chart illustrating various logic phases through which a system according to the invention may proceed.

[0013] FIG. 10B illustrates certain documents from FIG. 10A in layout space.

DETAILED DESCRIPTION

[0014] The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

[0015] In an embodiment of the invention, a system can have several aspects, and different embodiments need not implement all of them: 1) a module for creating an initial query, 2) a module for obtaining a set of candidate results satisfying the initial query, 3) a module for determining the distance or similarity between candidate results or a module for embedding the candidate results in a vector space, 4) a module for sub-selecting a discriminating set of candidate results, 5) a module for arranging candidate results in 2 dimensions, 6) a module for obtaining user input with regard to the candidate results, 7) a module for refining the search query to incorporate information regarding the user input encoded as geometric or distance constraints with respect to the embedding or distance measures of 3, 8) a module for iteratively obtaining a set of candidate results satisfying the initial query and the geometric or distance constraints accumulated from user input.

[0016] FIG. 8 is a block diagram of various components of an embodiment of a system according to the invention. It includes an embedding module 820 which calculates an embedding of the source documents into an embedding space, and writes the embedding information, in association with identification of the documents, into a document collection database 816. A user interaction module 822 receives queries and query refinement input from a user, and provides them to a query processing module 824. In one embodiment the user interaction module 822 includes a computer terminal, whereas in another embodiment it includes only certain network connection components through which the system communicates with an external computer terminal. The query processing module 824 interprets the queries as geometric constraints on the embedding space, and narrows the collec-

tion to develop a set of candidate documents which satisfy the geometric constraints. These are written into a candidate space database **826**. Candidate spaces as used herein are also embedding spaces, and for example may constitute a portion of the embedding space of collection **816**. In some embodiments query processing module may also perform a re-embedding of the candidate documents in embedding space. A discriminative selection module **828** then selects a discriminative set of the documents from the candidate space **826** and presents them to the user via user interaction module **822**. User interaction module **822** may then receive further refinement queries from the user, which are handled as above, or it may receive a user commit indication, in which case the system takes some action **830** with respect to the user's selected document such as opening the document for the user or engaging in further search refinement.

[0017] Note that in some embodiments the user refinement input may not require a further geometric constraint on the candidate space, but rather may involve only selection of a different discriminative set of documents from the existing candidate space **826** for presentation to the user. Also, in various embodiments, the candidate space database may not be implemented as a separate database, but rather may be combined in various ways with the document collection embedding space **816**. Candidate space may also be implied rather than physical in some embodiments.

[0018] FIG. 9 is a flow chart illustrating various logic phases through which a system according to the invention may proceed. Initially, in step **910**, a collection of digital documents (which, as used herein includes images, text, web-pages, catalog entries, and sections of documents) is embedded in embedding space. In step **912**, an initial query is optionally processed to yield an initial subset of digital documents satisfying the query results. (As used herein, the term "subset" refers only to a "proper" subset.) This may be a conventional text query, for example. In step **914**, the initial query is processed to yield a set of geometric constraints (possibly empty). In step **916** the geometric constraints are applied to the space of embedded digital documents or to the initial subset of digital documents to identify a set of candidate results. In step **918** a discriminative subset of documents is selected from the candidate results, and in step **920** the discriminative subset of documents is presented toward the user. In step **922** the user provides further input, for example by selecting or deselecting candidate results, by indicating a direction on the display with respect to one or more candidate results, or by providing a ranked order of the candidate results; or if satisfied with one of the candidate results, the user indicates to commit to that result. If the user input indicates further refinement, then the logic returns to step **914** to process the refinement query to yield further geometric constraints step **924**). If not, then in step **926** the system takes action with respect to the user-selected document.

[0019] Note that at least steps **910**, **912** and **914** can happen in any order. In one embodiment, step **910** occurs continuously in the background, separately from the remainder of the steps, and updates the document collection in embedding space asynchronously with the remainder of the steps. In general, the logic of FIG. 9, as well as all sequences and flow charts herein, can be implemented using processors programmed using computer programs stored in memory accessible to the computer systems and executable by the processors, by dedicated logic hardware, including field programmable integrated circuits, or by combinations of

dedicated logic hardware and computer programs. Each block in the flowchart or phase in a logic sequence describes logic that can be implemented in hardware or in software running on one or more computing processes executing on one or more computer systems. In one embodiment, each step of the flow chart or phase in a logic sequence illustrates or describes the function of a separate module of software. In another embodiment, the logic of the step is performed by software code routines which are distributed throughout more than one module. As with all flowcharts and logic sequences herein, it will be appreciated that many of the steps can be combined, performed in parallel or performed in a different sequence without affecting the functions achieved. In some cases, as the reader will appreciate, a re-arrangement of steps will achieve the same results only if certain other changes are made as well. In other cases, as the reader will appreciate, a re-arrangement of steps will achieve the same results only if certain conditions are satisfied. Furthermore, it will be appreciated that the flow charts and logic sequences herein show only aspects that are pertinent to an understanding of the invention, and it will be understood that in a specific embodiment, numerous additional steps for accomplishing other functions for that embodiment can be performed before, after and between those steps shown.

Embeddings of Digital Documents

[0020] The "embedding space", into which digital documents are embedded by embedding module **820** and in step **910**, as used herein is a geometric space within which documents are represented. In one embodiment the embedding space is a vector space, in which the features of a document define its "position" in the vector space relative to an origin. The position is typically represented as a vector from the origin to the document's position, and the space has a number of dimensions based on the number of coordinates in the vector. Vector spaces deal with vectors and the operations that may be performed on those vectors. In another embodiment the embedding space is a metric space, which does not have a concept of position, dimensions or an origin. Distances among documents in a metric space are maintained relative to each other, rather than relative to any particular origin. Metric spaces deal with objects combined with a distance between those objects and the operations that may be performed on those objects. For purposes of the current discussion these objects are significant in that there exist many efficient algorithms that operate on vector spaces and metric spaces. For example metric trees may be used to rapidly identify objects that are "close" to each other. In the discussion below we embed objects into vector spaces and/or metric spaces. In the context of a vector space this means that we define a function that maps objects to vectors in some vector space. In the context of a metric space it means that we define a metric (or distance) between those objects that allows us to treat the set of all such objects as a metric space. Note that vector spaces allow the use of a variety of standard measures of distance (divergence) including the Euclidean distance. Other embodiments can use other types of embedding spaces.

[0021] Let us first consider embedding in a vector space. To embed a document collection in a vector space we associate each document with a vector. The distance between two documents in such a space is then determined using standard measures of distance on vectors.

[0022] The goal of embedding digital documents in a vector space is to place intuitively similar documents close to

each other. There are many ways to achieve this. For example a common way of embedding text documents is to use a bag-of-words model. The bag of words model maintains a dictionary. Each word in the dictionary is given an integer index, for example, the word *aardvark* may be given the index 1, and the word *zebra* may be given the index 60,000. Each document is processed by counting the number of occurrences of each dictionary word in that document. A vector is created where the value at the i^{th} index is the count for the i^{th} dictionary word. Variants of this representation normalize the counts in various ways. Such an embedding captures information about the content and therefore the meaning of the documents. Text documents with similar word distributions are close to each other in this embedded space.

[0023] There are a wide variety of means by which documents may be embedded into a vector space. For example images may be processed to identify commonly occurring features using, e.g., scale invariant feature transforms (SIFT), which are then binned and used in a representation similar to the bag-of-words embedding described above. Of particular interest are embeddings created using deep neural networks, or other deep learning techniques. For example a neural network can learn an appropriate embedding by performing gradient descent against a measure of dimensionality reduction on a large set of training data. As another example, one could learn a kernel based on data and derive a distance based on that kernel. Likewise one may learn a distance directly. These approaches generally use large neural networks to map documents, words, or images to high dimensional vectors. Alternatively, one may learn an embedding using examples with algorithms such as Multi-Dimensional Scaling, or Stochastic Neighbor Embedding. An embedding into a vector space may also be defined implicitly via a kernel. In this case the explicit vectors may never be generated or used, rather the operations in the vectors space are carried out by performing kernel operations in the original space.

[0024] In many cases a very high dimensional space would be required to capture the intuitive relationships between documents. In some of these cases the required dimensionality may be reduced by choosing to embed the documents on a manifold (curved surface) in the space rather than to arbitrary locations.

[0025] Finally, it is worth noting that different embeddings may be appropriate on different subsets of the document collection. For example, it may be most effective to re-embed the candidate result sets at each iteration of the search procedure. In this way the subset may be re-embedded to capture the most important axes of variation or of interest in that subset.

[0026] To embed a digital document collection in a metric space requires associating that collection with a distance (or metric). Below we discuss a number of ways to define a distance between digital documents.

Distances and Similarities Between Digital Documents

[0027] “Distance” between two documents in embedding space “corresponds to” a predetermined measurement of dissimilarity among documents. Preferably it is a monotonic function of the measurement of dissimilarity. Typically it equals the measurement of dissimilarity. Example distances include the Manhattan distance, the Euclidean distance, and the Hamming distance.

[0028] There are a wide variety ways to measure the distance (or dissimilarity) between digital documents, and these

may be combined to produce new measures of distance. An important concept is that the intuitive relationships between digital documents may be captured via such a similarity or distance measure. For example, some useful distance measures place images containing the same person in the same place close to each other. Likewise, some useful measures place documents discussing the same topic close to each other. Of course there are many axes along which digital documents may be intuitively related, so that the set of all documents close (with respect to that distance) to a given document may be quite diverse. For example, a historical text describing the relationship between Anthony and Cleopatra may be similar to other historical texts, texts about Egypt, texts about Rome, movies about Anthony and Cleopatra, and love stories. Each of these types of differences constitutes a different axis relative to the original historical text.

[0029] Such distances may be defined in a variety of ways. One typical way is via embeddings into a vector space. Other ways include encoding the similarity via a Kernel. By associating a set of documents with a distance we are effectively embedding those documents into a metric space. Documents that are intuitively similar will be close in this metric space while those that are intuitively dissimilar will be far apart. Note further that kernels and distance functions may be learned. In fact, it may be useful to learn new distance functions on subsets of the documents at each iteration of the search procedure.

Database Organization

[0030] The databases used in an embodiment of the invention, such as databases **816** and **826**, may use commonly available means to store the data in, e.g., a relational database, a document store, a key value store, or other related technologies. In each case the original document contents (or pointers to them) may be stored and associated with their high dimensional representation, or a set of measures of distance relative to other documents.

[0031] In order to achieve scalable and fast search performance indexing structures are critical. When documents are embedded in a vector space indexes may be built using, e.g., kd-trees. When documents are associated with a distance metric and hence embedded in a metric space metric trees may be used.

[0032] The databases described herein are stored on one or more non-transitory computer readable media. As used herein, no distinction is intended between whether a database is disposed “on” or “in” a computer readable medium. Additionally, as used herein, the term “database” does not necessarily imply any unity of structure. For example, two or more separate databases, when considered together, still constitute a “database” as that term is used herein.

Initial Query

[0033] The initial query presented to the system in step **912** may be created and evaluated using a variety of standard techniques. For example, the query may be presented as a set of keywords entered via a keyboard or via speech; the query may be a natural language phrase, or sentence entered via a keyboard or via speech; or the query may be an audio signal, an image, a video, or a piece of text representing a prototype for which similar audio signals, images, videos, or text may be sought. A variety of means are known by which such an

initial query may be efficiently evaluated, e.g., searching a relational database, or using an inverted index.

[0034] The initial query may also be designed to simply return a random set of results.

[0035] Other interfaces for initial queries allow for faceted search. Faceted search provides a means for users to constrain a search along a set of axes. For example, the faceted search might provide a slider that allows users to constrain the range of acceptable prices.

Identifying Candidate Results

[0036] The search constraints created from an initial query and subsequent user input are used to identify a set of candidate results. This may be achieved using a variety of means. For example, the initial query may be performed against a relational database whereby the results are then embedded in a vector or metric space. These results may then be indexed using, e.g., a kd-tree or a metric tree and searched to identify candidates that satisfy both the initial query and the constraints. Alternatively, the initial query may also be converted to geometric constraints that are applied to the set of embedded documents. In this case the geometric representation of the constraints implied both by the initial query and the user input are combined and an appropriate index is used to identify embedded documents satisfying both sets of constraints.

[0037] As used herein, a “geometric constraint” on an embedding space is a constraint that is described formulaically in the embedding space, rather than only by cataloguing individual documents or document features to include or exclude. In a vector embedding space, for example, the constraint can be described in the form of a specified function which defines a hypersurface. Documents on one side of the hypersurface satisfy the constraint whereas documents on the other side do not. A hyperplane may be defined in terms of dot products or kernels and requires that $k(x,z) > 0$ for a fixed vector x and a candidate z . Likewise a conic constraint may require that $k(x,z) > c$ for some constant c . In a metric embedding space, the constraint can be described in the form of a function of, for example, distances between documents. Thus in a metric embedding space, a geometric constraint might take the form of ‘all documents within a specified distance from document X ’, for example, or ‘all documents whose distance to document A is less than its distance to document B ’. Geometric constraints also may be combined using set operations, e.g., union, intersection to define more complex geometric constraints. They also may be created by taking transformations of any of the example constraints discussed. For example, one may take a polynomial function of distances, e.g., $d(x,z)*d(x,z)+d(y,z)<d(w,z)$ for given documents x, y , and w , and only those documents z which satisfy the function are considered to satisfy the geometric constraint.

[0038] In various embodiments, constraints (either as indicated by the user or as converted to geometric constraints) may be “hard” or “soft”. Hard constraints are those which must be satisfied in the sense that solutions must satisfy the conditions of all hard constraints. Soft constraints are those which need not be satisfied but candidate solutions may be penalized for each soft constraint that they don’t satisfy. Solutions may be rejected in a particular embodiment if the accumulation of such penalties is too large. Constraints may be relaxed in some embodiments, for example hard constraints

may be converted to soft constraints by associating them with a penalty, and soft constraints may have their penalties reduced.

Discriminative Result Set

[0039] Search queries may be ambiguous, or underspecified and so the documents satisfying a query may be quite diverse. For example, if the initial query is for a “red dress” the results may be quite varied in terms of their length, neckline, sleeves, etc. This aspect of the module sub-selects a discriminating set of results. Intuitively the objective is to provide a set of results to the user such that selection or de-selection of those results provides the most informative feedback or constraints to the search algorithm. One may think of this step as identifying an “informative” set of results, or a “diverse” set of results, or a “discriminating” set of results. Discriminative selection module **828**, performing step **918**, selects a discriminative subset of results in any of a variety of ways.

[0040] In one embodiment, a subset of the results may be discriminative as it provides a diversity of different kinds of feedback that the user can select. Diverse images may be selected as in, e.g., van Leuken, et. al., “Visual Diversification of Image Search Results”, in WWW ’09 Proceedings of the 18th international conference on World wide web, pp. 341-350 (2009), incorporated by reference herein. This diverse set is selected in order to provide the user with a variety of ways in which to refine the query at the next iteration. There are a variety of ways in which such a set may be identified. For example, farthest first traversal may be performed which incrementally identifies the “most” diverse set of results. Farthest first traversal requires only a distance measure and does not require an embedding. Farthest first traversal may also be initialized with a set of results; subsequent results are then the most different from that initial set.

[0041] Other means for selecting discriminative subsets of candidate results include using an algorithm like PCA (principal component analysis) or kernel PCA to identify the key axes of variation in the complete set of results. The discriminative subset is then constructed to contain documents that lie at multiple points along those most discriminating axes.

[0042] Another means might consider the set of constraints that would results from the user selecting or deselecting a given document. This set of constraints may be considered in terms of the candidate results it would yield. A discriminative subset may be selected so that the sets of candidate results produced by selecting any of the documents in that discriminative subset are as different as possible.

[0043] As used herein, “discriminativeness” of a particular set of documents in a collection of documents is the least number of documents in the collection that are excluded as a result of user selection of any document in the set. That is, if user selection of different documents in the particular set results in excluding different numbers of documents in the collection, then the set’s “discriminativeness” is considered herein to be the least of those numbers. Note that either the discriminative set of documents, or the formula by which user selection of a document determines which documents are to be excluded, or both, should be chosen such that the union of the set of documents excluded by selecting any of the documents in a discriminative set equals the entire collection of documents.

[0044] Also as used herein, the “average discriminativeness” of a set of size n documents in a collection of docu-

ments, is the average, over all sets of size n documents in the collection of documents, of the discriminativeness of that set. Also as used herein, one particular set of documents can be “more discriminative” than another set of documents if the discriminativeness of the first set is greater than the discriminativeness of the second set.

[0045] Preferably the selection module **828**, performing step **918**, selects a set of $N1 > 1$ documents from the current candidate space **826**, which is more discriminative than the average discriminativeness of sets of size $N1$ documents in the candidate space. Even more preferably, selection module **828** performing step **918** selects a set which is at least as discriminative as all other sets of size $N1$ documents in the current candidate space.

Result Presentation

[0046] The aim of the discriminative results presentation to the user in step **920** by user interaction module **822**, is to provide the user with a framework in which to refine the query constraints.

[0047] For example the results may be presented as a two dimensional grid. Results should be placed on that grid in a way that allows the user to appreciate the underlying distances between those results (as defined using a distance measure or embedding). One way to do this would be to ensure that results that are far from each other with respect to the distance measure are also displayed far from each other on the grid. Another way would be to project the embedding space onto two dimensions for example using multidimensional scaling (MDS) (for example see: Jing Yang, et. al., Semantic image browser: Bridging information visualization with automated intelligent image analysis, Proc. IEEE Symposium on Visual Analytics Science and Technology (2006), incorporated herein by reference). Yet another way would be to sub-select axes in the embedding space and position results along those axes.

[0048] Other layouts contemplated include 2 dimensional organizations not on a grid (possibly including overlapping results), 3 dimensional organizations analogous to the 2 dimensional organizations. Multi-dimensional organizations analogous to the 2 and 3 dimensional organizations with the ability to rotate around one or more axes. In general an M -dimensional layout can be used, where $M > 1$. In embodiments in which the embedding space has dimensions, the number of dimensions in the presentation layout need not be the same as the number of dimensions in the embedding space. Yet other layouts include hierarchical organizations or graph based layouts.

[0049] The document placement in the layout space should be indicative of the relationship among the documents in embedding space. For example, the distance between documents in layout space should correspond (monotonically, if not linearly) with the distance between the same documents in embedding space. Also, if three documents are collinear in embedding space, advantageously they are placed collinearly in layout space as well. In particular, collinearity in layout space with a candidate document which the system identifies as the most likely target of the user’s query (referred to herein as the primary candidate document) indicates collinearity in the embedding space with the primary candidate document.

[0050] It will be appreciated, however, that embedding space typically has a very large number of dimensions, and in high dimensional spaces very few points are actually collinear. In an embodiment, therefore, documents presented

collinearly in layout space indicate only “substantial” collinearity in embedding space. As used herein, if the embedding space is such that each document has a position in the space (as for a vector space), then three documents are considered “substantially collinear” in embedding space if the largest angle of the triangle formed by the three documents in embedding space is greater than 160 degrees. If the embedding space is such that documents do not have a position in the embedding space, but they do have distances from each other (such as for a metric space), then as used herein, a group of three documents are considered collinear if the sum of the two smallest distances between pairs of the documents in the group in embedding space equals the largest distance between pairs of the documents in the group in embedding space. The three documents are considered “substantially collinear” if the sum of the two smallest distances exceeds the largest distance by no more than 10%. As used herein, “collinearity” and “substantial collinearity” do not include the trivial cases of coincidence or substantial coincidence.

User Input

[0051] User interaction module **822** provides the user with a user interface (UI) which allows the user to provide input in a variety of ways. For example, the user may click on a single result to select it, or may swipe in the direction of a single result to de-select it. Similarly, the user may select or deselect multiple results at a time. For example, this may be done using a toggle selector on each result. The user might also implicitly select a set of results by swiping in the direction of a result indicating a desire for results that are more like that result “in that direction”. In this case “in that direction” means that the differences between the primary result and the result being swiped should be magnified. That is, the next set of results should be more like the result being swiped and less like the “primary result”. This concept may be generalized by allowing the user to swipe “from” one result “to” another result. In this case new results should be more like the “to” result and less like the “from” result.

[0052] Additionally, the UI can provide the user with the ability (e.g., via a double-click, or a pinch) to specify that the next set of results should be more like a specific result than any of the other results displayed. This is analogous to zooming on a map. Conversely, the UI can provide the ability to specify that the next set of results should be like a particular selection but more diverse than the currently selected set of results.

[0053] The UI may also provide the ability for the user to specify that a different set of similarly diverse images be provided.

Refined Query

[0054] The system then incorporates the user’s input to create a refined query. The refined query includes information regarding the initial query and information derived from the iterative sequence of refinements made by the user so far. This refined query may be represented as a set of geometric constraints that focus subsequent results within a region of the embedding space. Likewise, it may be represented as a set of distance constraints whose intersection defines the refined candidate set of results. It may also be represented as a path through the set of all possible results.

[0055] For example, the refined query may include constraints that require subsequent results to be within a specified

distance of one of the selected candidate results. Or the refined query may include constraints that require subsequent results to be closer (with respect to the distance measure) to one candidate result than to another.

[0056] Users may specify incompatible constraints. A system according to the invention may have the ability to relax, tighten, remove, or modify constraints that it determines are inappropriate.

Identifying Next Results

[0057] Given a distance (dissimilarity measure) between documents to be searched, or an embedding of those documents into a vector space or onto a manifold there are a variety of data structures that may be used to index that document collection and hence allow for rapid search. Such data structures include metric trees, kd-trees, R-trees, universal B-trees, X-trees, ball trees, locality sensitive hashes, and inverted indexes.

[0058] The system uses a combination of such data structures to identify the next set of candidate results based on the refined query.

Learning Distances

[0059] User behavior data may be collected by a system according to the invention and used to improve or to specialize the search experience. In particular, many ways of expressing distances or similarities may be parameterized and those parameters may be fit. For example, a similarity defined using a linear combination of kernels may have the coefficients of that linear combination tuned based on user behavior data. In this way the system may adapt to individual (or community, or contextual) notions of similarity.

Some Illustrations

[0060] FIG. 3 illustrates a set of documents embedded in 2-dimensional space. Aspects of the invention envision embedding documents in spaces of large dimensionality, hence two dimensions is for illustration purposes only. The space 310 contains documents, e.g., 321, 322. Each pair of documents has a distance 330 between them.

[0061] FIG. 4 illustrates the set of documents from FIG. 3 in addition to a circular geometric constraint 410. Those documents inside the circle, e.g., 421, 422 are said to satisfy the constraint. Aspects of the invention express queries and user input in the form of such geometric constraints. The documents that satisfy the constraints are the current results of the query. As the user provides further input additional constraints may be added, or existing constraints may be added or removed.

[0062] FIG. 5 illustrates the set of documents from FIG. 3 in addition to a non-circular geometric constraint 510. Aspects of the invention envision geometric constraints of arbitrary shape, and unions, intersections and differences of such constraints.

[0063] FIG. 6 illustrates a means by which the circular constraint of FIG. 4 may be updated in response to user input. The original circular constraint 610 may be modified by increasing its radius to produce circular constraint 620, or by decreasing its radius to produce constraint 630. These modifications are done in response to user input. The set of documents satisfying these constraints will change as the constraints are modified thus reducing or expanding the set of images considered for display to the user.

[0064] FIG. 7 illustrates a means by which a discriminative subset of documents may be selected for presentation to the user. The documents highlighted, e.g., 711, 712, are distinct from each other and from the others contained in the circular constraint region.

[0065] FIG. 10A illustrates a set of documents in embedding space, in which query processing module 824 has narrowed the collection to those documents within the circle 1020, and has identified primary result document 1018. In addition, discriminative selection module 828 has selected documents 1010, 1012, 1014 and 1016 as the discriminative set to present to the user. It can be seen that in embedding space, documents 1012, 1018 and 1016 are substantially collinear, and that documents 1010, 1018 and 1014 are substantially collinear.

[0066] FIG. 10B illustrates how the system may present the set of documents in layout space. (The broken lines are implied, rather than visible.) The specific positions of the documents do not necessarily match those in embedding space, in part because dimensionality of the space has been reduced. However, documents which were substantially collinear in embedding space are collinear in layout space. In particular, if the broken lines in FIG. 10A represent dimensions in embedding space along which the candidate documents differ, the placement of the documents in layout space in FIG. 10B are indicative of those same dimensions. In addition, the relative distances among the documents along each of the lines of collinearity in layout space also are indicative of the relative distances in embedding space.

Advantages Over Prior Systems

[0067] Various embodiments of the approaches described herein may yield one or more of the following advantages over prior systems:

[0068] An embodiment need not be limited to a single fixed hierarchy of documents. More specifically, an embodiment need not require the explicit determination of a taxonomy by which the document collection is described. Nor does it require a clustering of documents into a static hierarchy. That is, the sequence of refinements that a user may perform need not be constrained to narrowing or broadening in some pre-defined taxonomy or hierarchy.

[0069] An embodiment can be extremely flexible and may be applied to images, text, audio, video, and many other kinds of data.

[0070] Intuitions about the relationships among documents are often easier to express using notions of similarity or distance between documents than using a taxonomy or tags.

[0071] Selecting and deselecting candidate results in a visual way is a more facile interface for performing search on a mobile device or a tablet.

[0072] Encoding query refinements in terms of geometric constraints allows for a more flexible user interaction. Specifically, in an embodiment, the user is not required to be familiar with a pre-defined tagging ontology, or with a query logic used to combine constraints. Furthermore, in an embodiment such geometric constraints can be more robust to errors in a feature tagging or annotation process.

[0073] The ability to incrementally refine a search is helpful to a productive user experience.

[0074] The use of a discriminative subset of candidate results makes more effective use of limited display space. The clutter on the display is minimized while simultaneously capturing a high proportion of the information available in the complete results set and providing a wide variety of options for the user to refine a query.

[0075] Given that distances, embeddings, and similarities may be machine learned, a system using this approach can provide the ability to specialize the search experience to individuals, groups, cultures, and document categories.

[0076] Compared to content based image retrieval (CBIR) techniques, an embodiment of the invention can be more amenable to incremental refinement of a search. Specifically, a user may take a photograph and use a CBIR system to identify related or highly similar photographs. However, if the user is dissatisfied with the results the CBIR system does not provide them with a way to refine search goals.

Example 1

[0077] One implementation allows users to search a personal photograph collection. Users are initially shown an arbitrary photograph (the primary result), e.g., the most recent photograph taken or viewed. This is displayed in the center of a 3x3 grid of photographs from the collection. Each of the photographs is selected to be close (defined below) to the primary result but different from each other along different axes relative to the primary result. For example, if the primary result is a photograph taken with family last week at home, then other photographs may be a) with the family last year at home, b) with the family last week outdoors, c) without the family last week at home, etc. In some situations, the system may place two photographs on opposite sides of the primary result which are along the same axis but differ from each other in their positions along that axis. For example, the photo placed on the left side may show family member A more prominently than in the primary result, while the photo placed on the right side may show family member A less prominently than in the primary result.

[0078] The user selects one of the 9 photographs which then becomes the primary result. This is then laid out in an updated 3x3 grid of photographs again “close” to it but different from each other.

[0079] If at any point the user double clicks on the primary result then the definition of “close” changes to a “smaller scale” (defined below). If the user uses a “pinch out” gesture then the definition of “close” changes to a “larger scale” and the result set is updated.

[0080] In this way a user may navigate a collection of photographs to find specific ones.

[0081] In this example photographs may be considered similar with respect to a number of criteria, including:

[0082] GPS location of the photograph

[0083] Time of the photograph

[0084] Color content of the photograph

[0085] Whether the photograph was taken indoors or outdoors

[0086] Whether there are people in the photograph

[0087] Who is in the photograph

[0088] Whether people in the photograph are happy or sad

[0089] The activity depicted in the photograph

[0090] The objects contained in the photograph

[0091] And many others. These criteria are captured into a numerical “distance”, or as a vector locating photographs in some space. In the latter case a standard notion of similarity or distance may be used, e.g., the dot product or Euclidean distance. In an embodiment, a normalization function can be applied in order that distances along different axes are comparable to each other.

[0092] As the user navigates a photo collection the “scale” at which the user is searching changes. This scale specifies how “close” the photos in the result set are to the primary result. More precisely all photos in the result set must have a “distance” less than some threshold. As the scale increases or decreases this threshold increases or decreases.

[0093] Considering this example with respect to the steps described above:

[0094] Embedding: For each photograph in a user’s collection of personal photographs a vector is produced that has indices corresponding to, e.g., the longitude, the latitude, the time of day, the day of week, the number of faces, whether a given activity is depicted, among many others.

[0095] Initial Query: In this case the initial query is empty, that is all photos are candidate results and the one presented to the user is arbitrary.

[0096] Initial Query as geometric constraints: The initial query produces an empty set of geometric constraints

[0097] The geometric constraints are applied to the set of embedded photographs to identify those that satisfy the constraints, i.e., the candidate results

[0098] A discriminative subset of 9 photographs is selected from the candidate results using farthest first traversal.

[0099] The 9 photographs are presented to the user in a 3x3 grid

[0100] The user selects one of the photographs to indicate a desire to see more photographs like that one.

[0101] The user selected photograph is processed to yield a new geometric constraint which can be represented as a sphere around the selected photograph in the embedding space. This new constraint is added to the current set of constraints. The combined constraint is the intersection of spheres around all photographs selected so far.

Example 2

[0102] Another implementation looks at searching for accessories (apparel, furniture, apartments, jewelry, etc). In this implementation the user searches using text, speech, or with a prototype image as an initial query. For example, a user searches for “brown purse” using text entry. The search engine responds by identifying a diverse set of possible results, e.g., purses of various kinds and various shades of brown. These results are laid out in a 2 dimensional arrangement (for example a grid), whereby more similar results are positioned closer to each other and more different results are positioned relatively far from each other. The user then selects one or more images, for example using radio buttons. The image selections are then used by the search engine to define a “search direction” or a vector in the embedding space along which further results may be obtained.

[0103] Considering this example with respect to the steps described above:

[0104] Embedding: For each entry in an accessories catalog a vector is produced using deep learning techniques trained to differentiate accessories.

[0105] Initial Query: In this case the initial query is a textual search that narrows further results to be within a portion of the full catalog. This restricted is the set of initial candidate results.

[0106] Initial Query as geometric constraints: The initial query produces an empty set of geometric constraints

[0107] The geometric constraints are applied to the set of embedded accessories in the restricted set (i.e., the initial candidate results) to identify those that satisfy the constraints, i.e., the candidate results

[0108] A diverse subset of 9 catalog entries is selected from the candidate results using farthest first traversal.

[0109] The 9 catalog entries are presented to the user in a 3x3 grid

[0110] The user selects one of the catalog entries to indicate a desire to see more accessories like that one.

[0111] The user selected accessory is processed to yield a new geometric constraint which can be represented as a sphere around the selected accessory in the embedding space. This new constraint is added to the current set of constraints. The combined constraint is the intersection of spheres around all accessories selected so far.

Some Variations

[0112] Documents are encoded in an embedding space such as a vector space or metric space (via a distance). Searches proceed as a sequence of query refinements. Query refinements are encoded as geometric constraints over the vector space or metric space. Discriminative candidate results are displayed to provide the user with the ability to add discriminative constraints. User inputs, e.g., selecting or deselecting results, are encoded as geometric constraints.

[0113] Variations of this approach will be apparent to the reader. For example the documents may be embedded after the initial query is process and only those documents satisfying the query may be embedded. Similarly, the documents may be re-embedded using a different embedding at any point in the process. In this case, the geometric constraints would be re-interpreted in the new embedding.

[0114] The geometric constraints may be augmented at any point with non-geometric constraints. In this case the candidate results can be filtered in a straightforward way to select only those satisfying the non-geometric constraints. In this way the interaction can be augmented with faceted search, text, or speech inputs. At each iteration of the process the geometric constraints can be managed together with a set of non-geometric constraints.

[0115] An example implementation may proceed through these steps:

[0116] 1. Obtaining a collection of prototype images (at least 1) from a user;

[0117] 2. Identifying all images in the collection with a distance less than a threshold T from the prototype images;

[0118] 3. Identifying a discriminative subset of the images collected in (2);

[0119] 4. Presenting the discriminative subset of images to the user in a 2-dimensional layout;

[0120] 5. If the user is satisfied with one of the presented images, receiving an indication of such satisfaction and taking desired action with respect to the selected image;

[0121] 6. If the user is not yet satisfied, obtaining from the user a selection of the presented images more like the desired result;

[0122] 7. Producing a revised collection of prototype images;

[0123] 8. Reducing threshold T;

[0124] 9. Goto 2.

[0125] The above method may be viewed either from the viewpoint of the user interacting with a computer system, or the viewpoint of a computer system interacting with a user, or both.

[0126] The concept may be generalized to refer to “digital documents” rather than images, where digital documents include audio, video, text, html, multimedia documents and product listings in a digital catalog, in addition to images.

[0127] The concept may also be generalized so that the initial collection obtained at step 1 is obtained as the result of the user performing a search (query) within another information retrieval system or search engine.

[0128] The concept may also be generalized so that rather than reducing the threshold at step 8, the user interface provides for the ability to decrease or increase the threshold or leave it unchanged.

[0129] The concept may also be generalized so that at steps 1, and 6 there are two collections of prototype images and at step 2 the system identifies images with distance less than threshold T1 of the first set, and greater than T2 of the second set.

[0130] The concept may also be generalized so that at one iteration of step 6 the user selects image(s) along a first subset of at least one axis, and at another iteration of step 6 the user selects image(s) along a second subset of at least one axis, where the second subset of axes contains at least one axis not included in the first subset of axes.

Computer Environments

[0131] FIG. 1 illustrates an example environment in which aspects of the invention may be implemented. The system includes a user computer 110 and a server computer 112, connected to each other via a network 114 such as the Internet. The server computer 112 has accessibly thereto the database 816 identifying documents in association with embedding information, such as relative distances and/or their positions in a vector space. The user computer 110 also in various embodiments may or may not have accessibly thereto a database 118 identifying the same information.

[0132] Initially, embedding module 820 (which may for example be the server computer 112 or a separate computer system or a process running on such a computer) analyzes a collection of documents to extract embedding information about the documents. For example, if the documents are photographs, the embedding module 820 may include a neural network and may use deep learning to derive embedding image information from the photographs.

[0133] Alternatively, embedding module 820 may derive a library of image classifications (axes on which a given photograph may be placed), each in association with an algorithm for recognizing in a given photograph whether (or with what probability) the given photograph satisfies that classification. Then the embedding module 820 may apply its pre-developed library to a smaller set of newly provided photographs, such

as the photos currently on the user computer **110**, in order to determine embedding information applicable to each photograph. Either way, the embedding module **820** writes into the database **816** the identifications of the collection of documents that the user may search, each in association with its embedding information.

[0134] In yet another embodiment, the embedding information that embedding module **820** writes into database **816** may be provided from an external source, or entered manually.

[0135] The iterative identification steps described above can be implemented in a number of different ways. In one embodiment, all computation takes place on the server computer **112**, as the user iteratively searches for a desired document. The user, operating the user computer **110**, sees all results only by way of a browser. In this embodiment, it is not necessary that the user computer **110** have the document collection database **118** accessibly thereto. In another embodiment, the server computer **112** transmits its entire database **118** of documents in embedding space (or a subset of that database) to the user computer **110**, which writes it into its own database **118**. All computation takes place on the user computer **110** in such an embodiment, as the user iteratively searches for a desired document. Many other arrangements are possible as well.

Computer Hardware

[0136] FIG. 2 is a simplified block diagram of a computer system **210** that can be used to implement software incorporating aspects of the present invention. The drawing represents both an embodiment of user computer **110** and server computer **112**. While the above-described methods indicate individual logic steps or modules for carrying out specified operations, it will be appreciated that each step or module actually causes the computer system **210** to operate in the specified manner.

[0137] Computer system **210** typically includes a processor subsystem **214** which communicates with a number of peripheral devices via bus subsystem **212**. These peripheral devices may include a storage subsystem **224**, comprising a memory subsystem **226** and a file storage subsystem **228**, user interface input devices **222**, user interface output devices **220**, and a network interface subsystem **216**. The input and output devices allow user interaction with computer system **210**. Network interface subsystem **216** provides an interface to outside networks, including an interface to communication network **218**, and is coupled via communication network **218** to corresponding interface devices in other computer systems. Communication network **218** may comprise many interconnected computer systems and communication links. These communication links may be wireline links, optical links, wireless links, or any other mechanisms for communication of information, but typically it is an IP-based communication network. While in one embodiment, communication network **218** is the Internet, in other embodiments, communication network **218** may be any suitable computer network.

[0138] The physical hardware component of network interfaces are sometimes referred to as network interface cards (NICs), although they need not be in the form of cards; for instance they could be in the form of integrated circuits (ICs) and connectors fitted directly onto a motherboard, or in the form of macrocells fabricated on a single integrated circuit chip with other components of the computer system.

[0139] User interface input devices **222** may include a keyboard, pointing devices such as a mouse, trackball, touchpad, or graphics tablet, a scanner, a touch screen incorporated into the display, audio input devices such as voice recognition systems, microphones, and other types of input devices. In general, use of the term “input device” is intended to include all possible types of devices and ways to input information into computer system **210** or onto computer network **218**. It is by way of input devices **222** that the user provides queries and query refinements to the system.

[0140] User interface output devices **220** may include a display subsystem, a printer, a fax machine, or non-visual displays such as audio output devices. The display subsystem may include a cathode ray tube (CRT), a flat-panel device such as a liquid crystal display (LCD), a projection device, or some other mechanism for creating a visible image. The display subsystem may also provide non-visual display such as via audio output devices. In general, use of the term “output device” is intended to include all possible types of devices and ways to output information from computer system **210** to the user or to another machine or computer system. It is by way of output devices **220** that the system presents query result layouts toward the user.

[0141] Storage subsystem **224** stores the basic programming and data constructs that provide the functionality of certain embodiments of the present invention. For example, the various modules implementing the functionality of certain embodiments of the invention may be stored in storage subsystem **224**. These software modules are generally executed by processor subsystem **214**.

[0142] Memory subsystem **226** typically includes a number of memories including a main random access memory (RAM) **230** for storage of instructions and data during program execution and a read only memory (ROM) **232** in which fixed instructions are stored. File storage subsystem **228** provides persistent storage for program and data files, and may include a hard disk drive, a floppy disk drive along with associated removable media, a CD ROM drive, an optical drive, or removable media cartridges. The databases and modules implementing the functionality of certain embodiments of the invention may have been provided on a computer readable medium such as one or more CD-ROMs, and may be stored by file storage subsystem **228**. The host memory **226** contains, among other things, computer instructions which, when executed by the processor subsystem **214**, cause the computer system to operate or perform functions as described herein. As used herein, processes and software that are said to run in or on “the host” or “the computer”, execute on the processor subsystem **214** in response to computer instructions and data in the host memory subsystem **226** including any other local or remote storage for such instructions and data.

[0143] Bus subsystem **212** provides a mechanism by which the various components and subsystems of computer system **210** communicate with each other as intended. Although bus subsystem **212** is shown schematically as a single bus, alternative embodiments of the bus subsystem may use multiple busses.

[0144] Computer system **210** itself can be of varying types including a personal computer, a portable computer, a workstation, a computer terminal, a network computer, a television, a mainframe, a server farm, or any other data processing system or user device. In particular, it is envisaged that user computer **110** may be a hand-held device such as a tablet

computer or a smart-phone. Due to the ever-changing nature of computers and networks, the description of computer system 210 depicted in FIG. 2 is intended only as a specific example for purposes of illustrating the preferred embodiments of the present invention. Many other configurations of computer system 210 are possible having more or less components than the computer system depicted in FIG. 2.

[0145] While the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes described herein are capable of being distributed in the form of a computer readable medium of instructions and data and that the invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. As used herein, a computer readable medium is one on which information can be stored and read by a computer system. Examples include a floppy disk, a hard disk drive, a RAM, a CD, a DVD, flash memory, a USB drive, and so on. The computer readable medium may store information in coded formats that are decoded for actual use in a particular data processing system. A single computer readable medium, as the term is used herein, may also include more than one physical item, such as a plurality of CD-ROMs or a plurality of segments of RAM, or a combination of several different kinds of media. As used herein, the term does not include mere time varying signals in which the information is encoded in the way the signal varies over time.

[0146] As used herein, a given event or value is “responsive” to a predecessor event or value if the predecessor event or value influenced the given event or value. If there is an intervening processing element, step or time period, the given event or value can still be “responsive” to the predecessor event or value. If the intervening processing element or step combines more than one event or value, the signal output of the processing element or step is considered “responsive” to each of the event or value inputs. If the given event or value is the same as the predecessor event or value, this is merely a degenerate case in which the given event or value is still considered to be “responsive” to the predecessor event or value. “Dependency” of a given event or value upon another event or value is defined similarly.

[0147] As used herein, the “identification” of an item of information does not necessarily require the direct specification of that item of information. Information can be “identified” in a field by simply referring to the actual information through one or more layers of indirection, or by identifying one or more items of different information which are together sufficient to determine the actual item of information. In addition, the term “indicate” is used herein to mean the same as “identify”.

[0148] The applicant hereby discloses in isolation each individual feature described herein and any combination of two or more such features, to the extent that such features or combinations are capable of being carried out based on the present specification as a whole in light of the common general knowledge of a person skilled in the art, irrespective of whether such features or combinations of features solve any problems disclosed herein, and without limitation to the scope of the claims. The applicant indicates that aspects of the present invention may consist of any such feature or combination of features. In view of the foregoing description it will be evident to a person skilled in the art that various modifications may be made within the scope of the invention.

[0149] The foregoing description of preferred embodiments of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Obviously, many modifications and variations will be apparent to practitioners skilled in this art. In particular, and without limitation, any and all variations described, suggested or incorporated by reference in the Background section of this patent application are specifically incorporated by reference into the description herein of embodiments of the invention. In addition, any and all variations described, suggested or incorporated by reference herein with respect to any one embodiment are also to be considered taught with respect to all other embodiments. The embodiments described herein were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention for various embodiments and with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents.

1. A method for user identification of a desired document, comprising:

providing, accessibly to a computer system, a database identifying a first plurality of documents in an embedding space, the database identifying a distance between each pair of the documents in the embedding space corresponding to a predetermined measure of dissimilarity between the pair of documents;

in dependence upon a user query:

the computer system geometrically constraining the embedding space to develop a first candidate space, and

identifying toward the user a first set of $N1 > 1$ candidate documents from the first candidate space, the first set of candidate documents having fewer documents than the first candidate space and being more discriminative than the average discriminativeness of set size $N1$ documents in the first candidate space; and

taking action in response to user selection of a document identified toward the user.

2. The method of claim 1, wherein the set of candidate documents is at least as discriminative as all other sets of size $N1$ documents in the first candidate space.

3. The method of claim 1, wherein taking action comprises: the computer system geometrically constraining the first candidate space to develop a second candidate space, the second candidate space being a geometrically constrained portion of the first candidate space determined in dependence upon user selection of a subset of the documents in the first set of candidate documents;

identifying toward the user a second set of $N2 > 1$ candidate documents from the second candidate space, the second set of candidate documents having fewer documents than the second candidate space and being more discriminative than the average discriminativeness of set size $N2$ documents in the second candidate space; and

taking action in response to user selection of a document from the second set of candidate documents.

4. The method of claim 3, wherein the second candidate space is determined so as to include more documents which are similar to the user selected subset of documents than the

number of documents similar to the documents in the first set of candidate documents which are not in the user selected subset.

5. The method of claim 1, wherein the embedding space comprises a multi-dimensional vector space, each of the dimensions in the vector space defining a different basis on which documents can differ from each other,

and wherein the database identifies a position for each of the documents in the multi-dimensional vector space.

6. The method of claim 5, wherein taking action comprises: a computer system identifying toward the user a second set of $N2 > 1$ candidate documents from the first candidate space, at least one pair of documents in the second set of candidate documents having different coordinates along a second dimension in the multi-dimensional vector space and being more discriminative than the average discriminativeness of set size $N2$ documents in the first candidate space, wherein the second dimension differs from all the dimensions of the multi-dimensional vector space along which documents in the first set of candidate documents differ from each other; and

taking action in response to user selection of a document from the second set of candidate documents.

7. The method of claim 5, wherein identifying toward the user a first set of candidate documents from the first candidate space comprises presenting toward the user an M -dimensional layout, $M > 1$, identifying the first set of candidate documents, the placement of candidate documents in the layout being indicative of dimensions in the first candidate space along which the candidate documents differ.

8. The method of claim 7, wherein the placement of at least some candidate documents in the layout relative to each is further indicative of the distances among those candidate documents in the first candidate space.

9. The method of claim 5, wherein providing a database comprises:

determining, from the first plurality of documents, a first plurality of dimensions along which at least some of the documents in the first plurality of documents differ;

a computer system determining a position in each dimension in the first plurality of dimensions, for each of the documents in the first plurality of documents; and

a computer system associating in the database each document in the first plurality of documents with its position in each dimension in the first plurality of dimensions.

10. The method of claim 1, wherein identifying toward the user a first set of candidate documents from the first candidate space comprises presenting toward the user an M -dimensional layout, $M > 1$, identifying the first set of candidate documents, the placement of at least some candidate documents in the layout being indicative of the distances among the at least some candidate documents in the first candidate space.

11. The method of claim 10, further comprising identifying toward the user a primary candidate document,

wherein at least one pair of the documents in the set of candidate documents which are substantially collinear with the primary candidate document in the first candidate space appear in the layout collinearly with the primary candidate document.

12. The method of claim 1, further comprising identifying toward the user a primary candidate document,

wherein identifying toward the user a first set of candidate documents from the first candidate space comprises pre-

sending toward the user an M -dimensional layout, $M > 1$, identifying the first set of candidate documents,

and wherein the layout is such that collinearity with the primary candidate document of documents in the layout indicates substantial collinearity with the primary candidate document of documents in the embedding space.

13. The method of claim 1, wherein identifying toward the user a first set of candidate documents from the first candidate space comprises presenting toward the user an M -dimensional layout, $M > 1$, identifying the first plurality of candidate documents, the placement of at least some candidate documents in the layout relative to each other being indicative of the distances among the at least some candidate documents in the first candidate space.

14. A system for user identification of a desired document, comprising:

a processor; and

a computer readable medium coupled to the processor, the computer readable medium having stored thereon in a non-transitory manner a plurality of software code portions defining logic for:

a first module to provide in a computer readable medium in a non-transitory manner a database identifying a first plurality of documents in an embedding space, the database identifying a distance between each pair of the documents in the embedding space corresponding to a predetermined measure of dissimilarity between the pair of documents, and

a second module to, in dependence upon a user query: geometrically constrain the embedding space to develop a first candidate space; and

identify toward the user a first set of $N1 > 1$ candidate documents from the first candidate space, the first set of candidate having fewer documents than the first candidate space and documents being more discriminative than the average discriminativeness of set size $N1$ documents in the first candidate space.

15. The system of claim 14, wherein the set of candidate documents is at least as discriminative as all other sets of size $N1$ documents in the first candidate space.

16. The system of claim 14, wherein the plurality of software code portions further define logic for:

a module to geometrically constrain the first candidate space to develop a second candidate space, the second candidate space being a geometrically constrained portion of the first candidate space determined in dependence upon user selection of a subset of the documents in the first set of candidate documents; and

a module to identify toward the user a second set of $N2 > 1$ candidate documents from the second candidate space, the second set of candidate documents having fewer documents than the second candidate space and being more discriminative than the average discriminativeness of set size $N2$ documents in the second candidate space.

17. The system of claim 16, wherein the second candidate space is determined so as to include more documents which are similar to the user selected subset of documents than the number of documents similar to the documents in the first set of candidate documents which are not in the user selected subset.

18. The system of claim 14, wherein the embedding space comprises a multi-dimensional vector space, each of the

dimensions in the vector space defining a different basis on which documents can differ from each other,

and wherein the database identifies a position for each of the documents in the multi-dimensional vector space.

19. The system of claim **18**, wherein the plurality of software code portions further define logic for a module to identify toward the user a second set of $N_2 > 1$ candidate documents from the first candidate space, at least one pair of documents in the second set of candidate documents having different coordinates along a second dimension in the multi-dimensional vector space and being more discriminative than the average discriminativeness of set size 2 documents in the first candidate space, wherein the second dimension differs from all the dimensions of the multi-dimensional vector space along which documents in the first set of candidate documents differ from each other.

20. The system of claim **18**, wherein identifying toward the user a first set of candidate documents from the first candidate space comprises presenting toward the user an M-dimensional layout, $M > 1$, identifying the first set of candidate documents, the placement of candidate documents in the layout being indicative of dimensions in the first candidate space along which the candidate documents differ.

21. The system of claim **20**, wherein the placement of at least some candidate documents in the layout relative to each is further indicative of the distances among those candidate documents in the first candidate space.

22. The system of claim **18**, wherein the module to provide a database comprises a plurality of software code portions defining logic for:

- a module to determine, from the first plurality of documents, a first plurality of dimensions along which at least some of the documents in the first plurality of documents differ;
- a module to determine a position in each dimension in the first plurality of dimensions, for each of the documents in the first plurality of documents; and
- a module to associate in the database each document in the first plurality of documents with its position in each dimension in the first plurality of dimensions.

23. The system of claim **14**, wherein identifying toward the user a first set of candidate documents from the first candidate space comprises presenting toward the user an M-dimensional layout, $M > 1$, identifying the first set of candidate documents, the placement of at least some candidate documents in the layout being indicative of the distances among the at least some candidate documents in the first candidate space.

24. The system of claim **23**, wherein the second module further identifies toward the user a primary candidate document,

wherein at least one pair of the documents in the set of candidate documents which are substantially collinear with the primary candidate document in the first candidate space appear in the layout collinearly with the primary candidate document.

25. The system of claim **14**, wherein the second module further identifies toward the user a primary candidate document,

wherein identifying toward the user a first set of candidate documents from the first candidate space comprises presenting toward the user an M-dimensional layout, $M > 1$, identifying the first set of candidate documents, and wherein the layout is such that collinearity with the primary candidate document of documents in the layout indicates substantial collinearity with the primary candidate document of documents in the embedding space.

26. The system of claim **1**, wherein identifying toward the user a first set of candidate documents from the first candidate space comprises presenting toward the user an M-dimensional layout, $M > 1$, identifying the first plurality of candidate documents, the placement of at least some candidate documents in the layout relative to each other being indicative of the distances among the at least some candidate documents in the first candidate space.

27. A computer readable medium, for use with a database identifying a first plurality of documents in an embedding space, the database identifying a distance between each pair of the documents in the embedding space corresponding to a predetermined measure of dissimilarity between the pair of documents, the computer readable medium having stored thereon in a non-transitory manner a plurality of software code portions defining logic for, in dependence upon a user query:

geometrically constraining the embedding space to develop a first candidate space, and
identifying toward the user a first set of $N_1 > 1$ candidate documents from the first candidate space, the first set of candidate documents having fewer documents than the first candidate space and being more discriminative than the average discriminativeness of set size N_1 documents in the first candidate space.

28. A system for user identification of a desired document, comprising:

means providing access to a database identifying a first plurality of documents in an embedding space, the database identifying a distance between each pair of the documents in the embedding space corresponding to a predetermined measure of dissimilarity between the pair of documents; and

means for, in response to a user query:

geometrically constraining the embedding space in dependence upon the query to develop a first candidate space, and

identifying toward the user a first set of $N_1 > 1$ candidate documents from the first candidate space, the first set of candidate documents having fewer documents than the first candidate space and being more discriminative than the average discriminativeness of set size N_1 documents in the first candidate space; and

means for receiving user input for further refinement of the set of candidate documents.

* * * * *