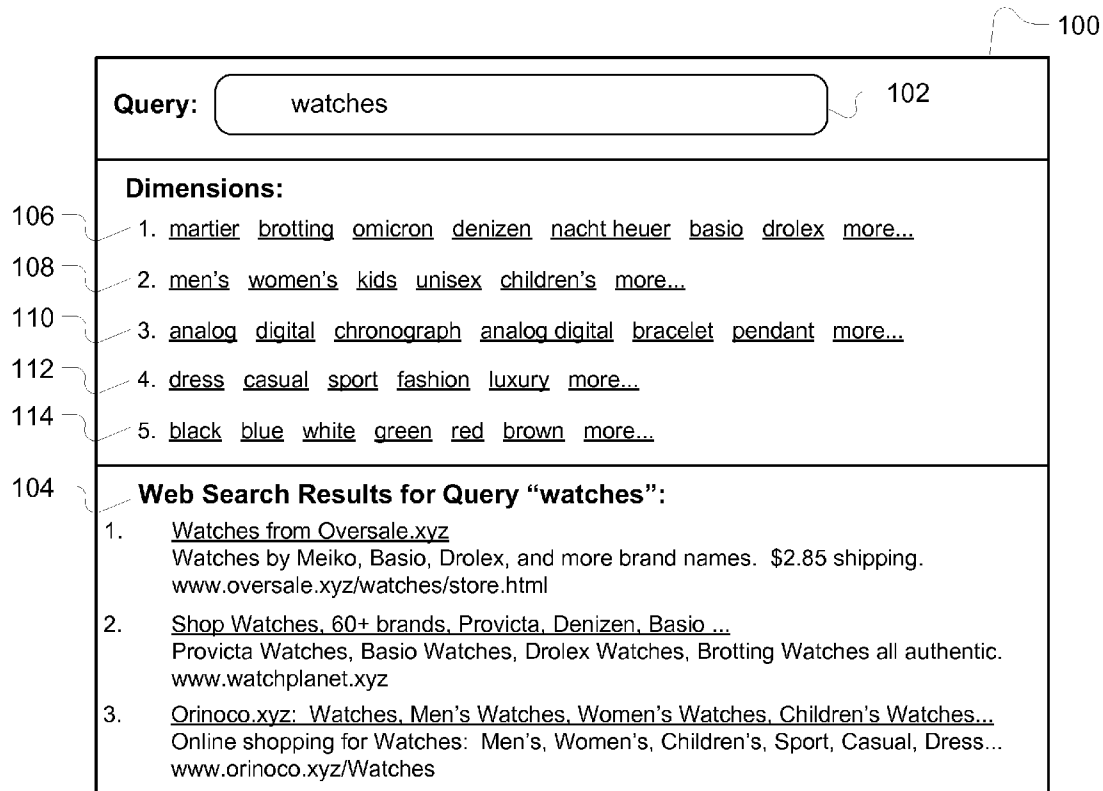




US 20130173605A1

(19) **United States**(12) **Patent Application Publication****Dou et al.**(10) **Pub. No.: US 2013/0173605 A1**(43) **Pub. Date: Jul. 4, 2013**(54) **EXTRACTING QUERY DIMENSIONS FROM  
SEARCH RESULTS**(52) **U.S. Cl.**  
USPC ..... **707/723; 707/E17.014**(75) Inventors: **Zhicheng Dou**, Beijing (CN); **Ruihua  
Song**, Beijing (CN); **Ji-Rong Wen**,  
Beijing (CN)(57) **ABSTRACT**(73) Assignee: **MICROSOFT CORPORATION**,  
Redmond, WA (US)

Techniques are described for automatically mining query dimensions from web pages resulting from execution of a search query. Lists of items such as words, terms, or phrases are extracted from the web pages based on the recognition of free text, metadata tag, or repeated region patterns within the web page text. Extracted item lists are weighted according to document matching and/or inverse document frequency, and item lists are clustered based on shared or similar items within the lists to generate query dimensions. The generated query dimensions, and the items within each query dimension, are ranked according to quality, and high-quality query dimensions are provided for display alongside top search results.

(21) Appl. No.: **13/343,621**(22) Filed: **Jan. 4, 2012****Publication Classification**(51) **Int. Cl.**  
**G06F 17/30** (2006.01)

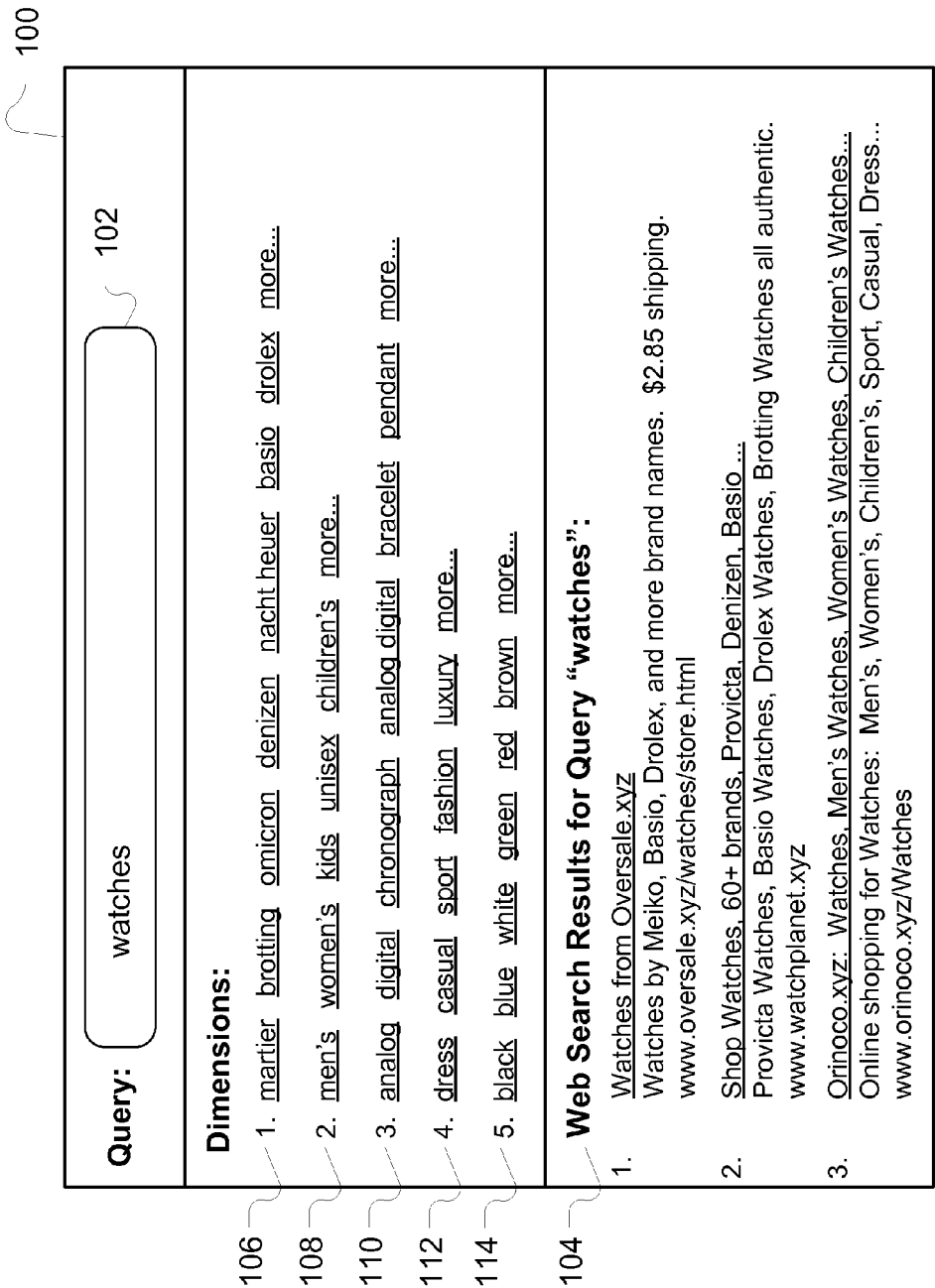


FIG. 1

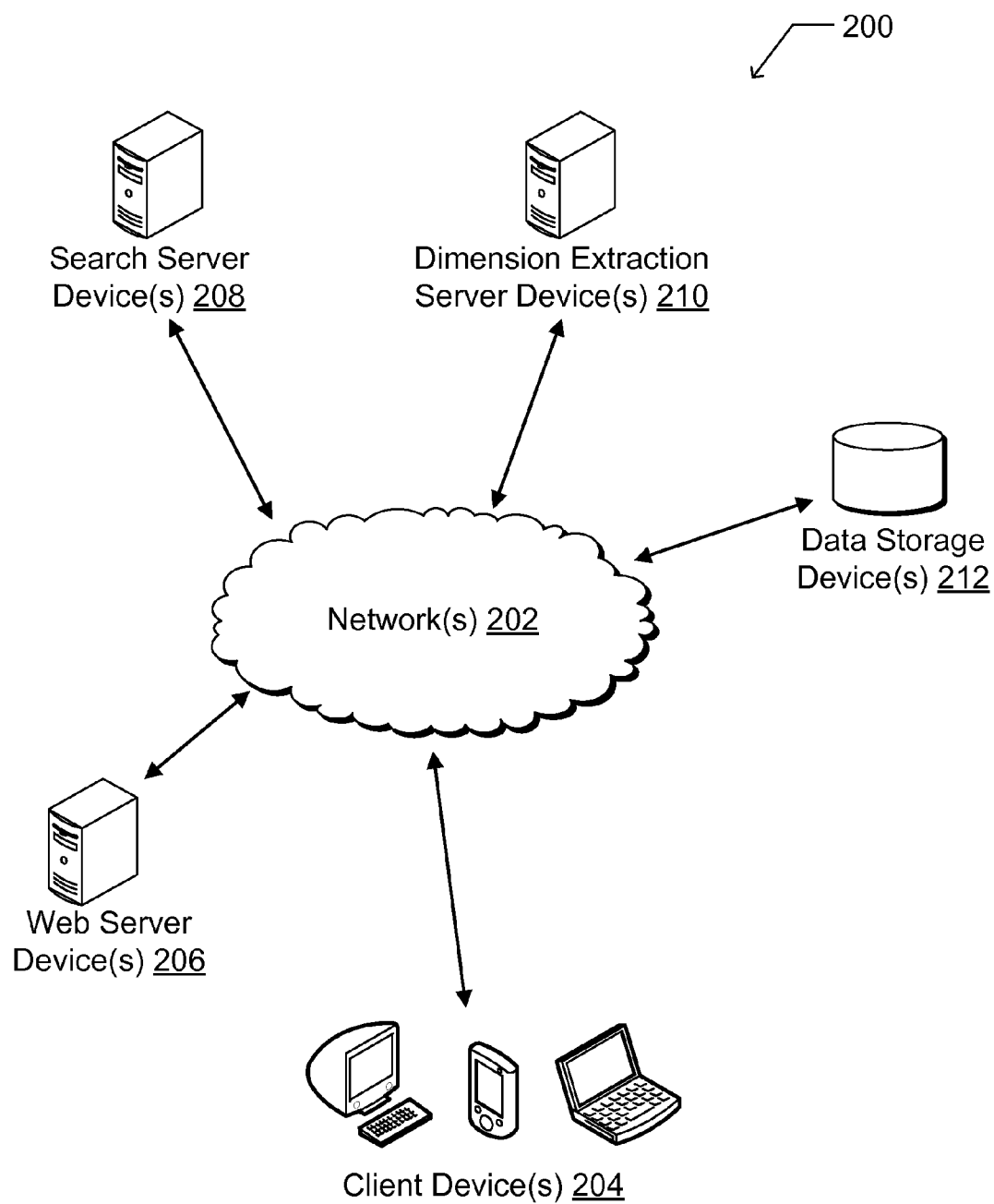


FIG. 2

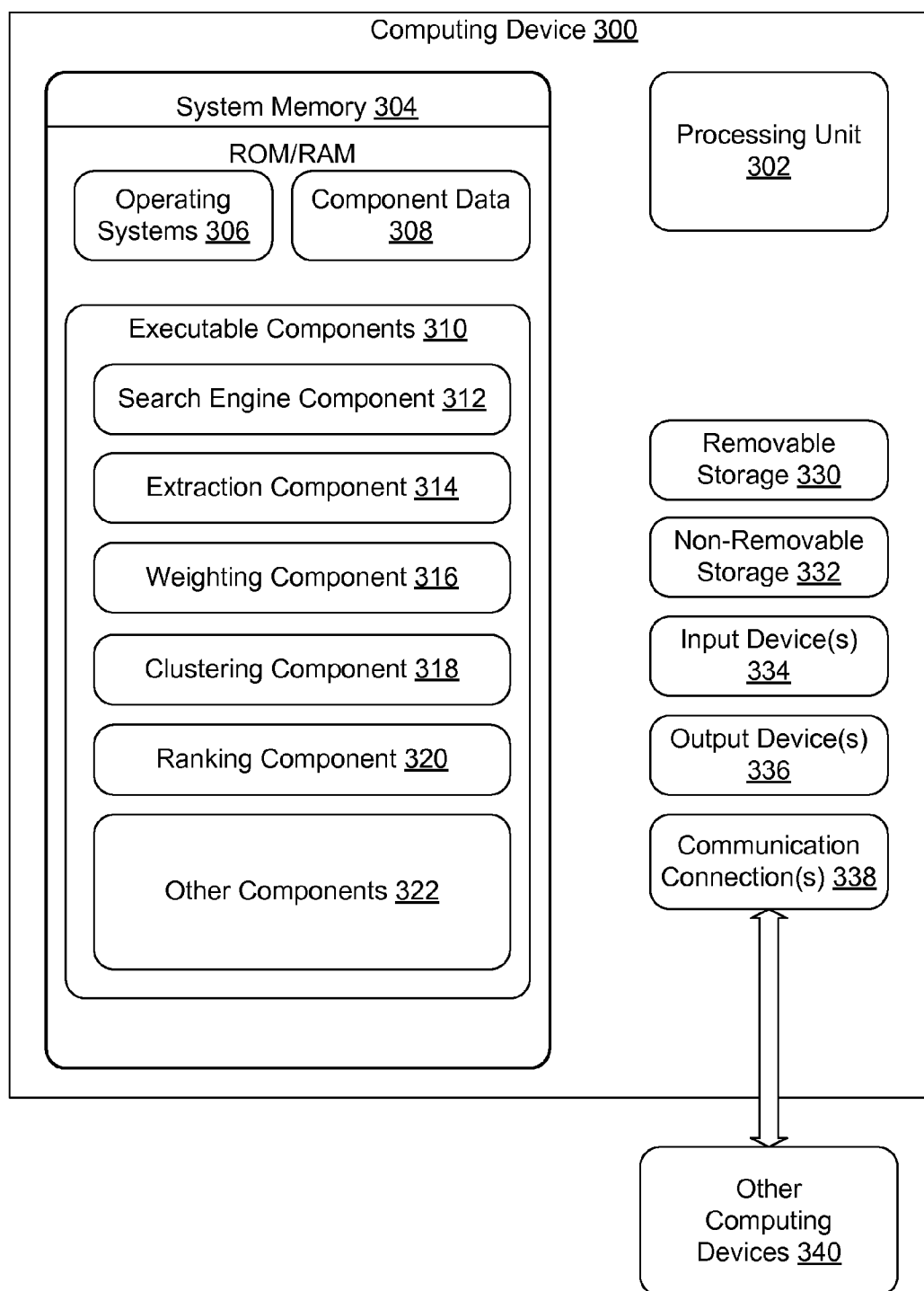


FIG. 3

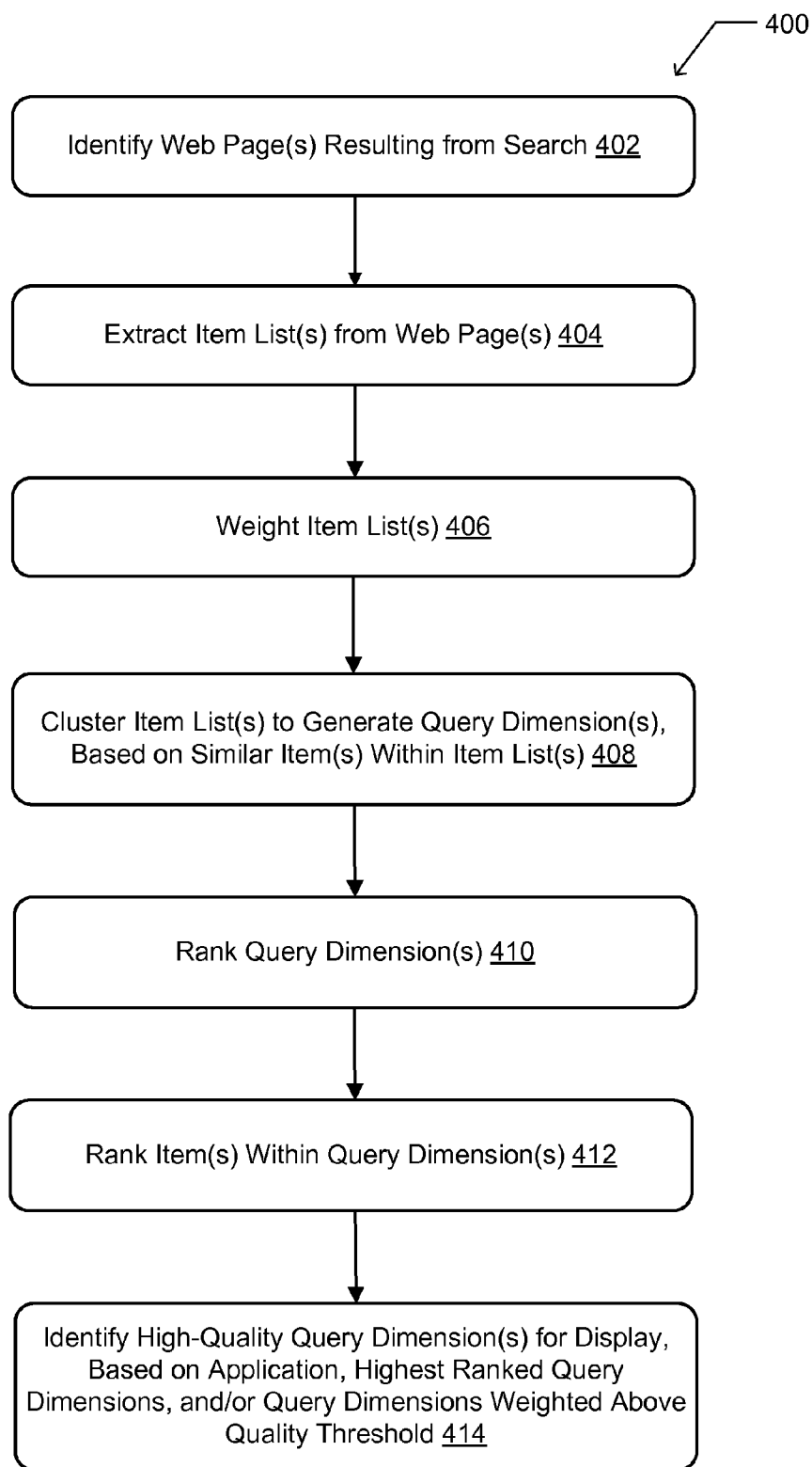
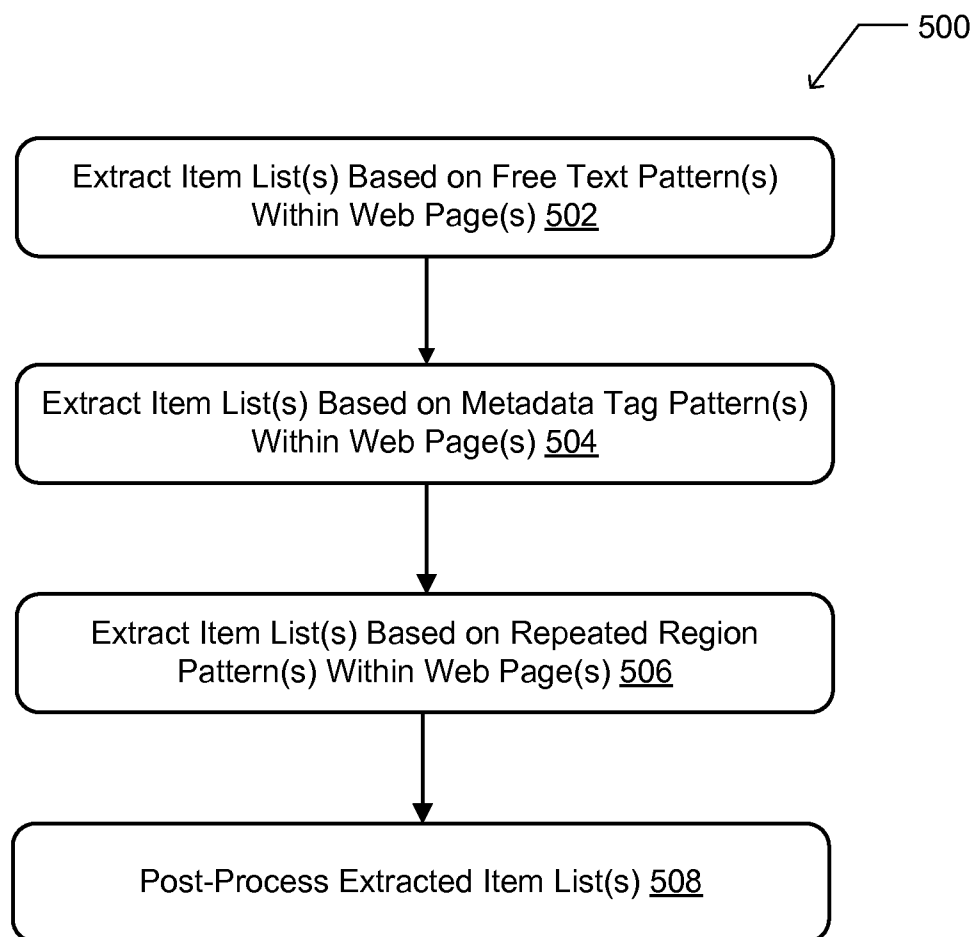


FIG. 4

**FIG. 5**

600

OVERSALE.XYZ

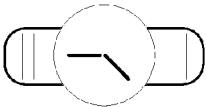
6 Months!  
Pay later

MEN'S WATCHES   WOMEN'S WATCHES   JEWELRY   SUNGLASSES   ACCESSORIES   TOP SELLERS

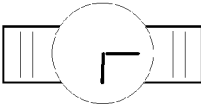
Accessories Watches (2)  
Basio Watches (11)  
Brotting Watches (21)  
Denizen Watches (8)  
Drolex Watches (3)  
Martier Watches (13)  
Nacht Heuer Watches (4)  
Omicron Watches (4)

Fall Savings!  
Prices Low!  
Shop Now!

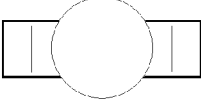
Shop  
Dutch Legend  
Save up to 78%



Shop  
Provicta  
Save up to 68%



Shop  
Jean Pepin/Geneve  
Save up to 81%



Shop  
Aquamarine  
Save up to 65%

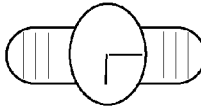


FIG. 6A

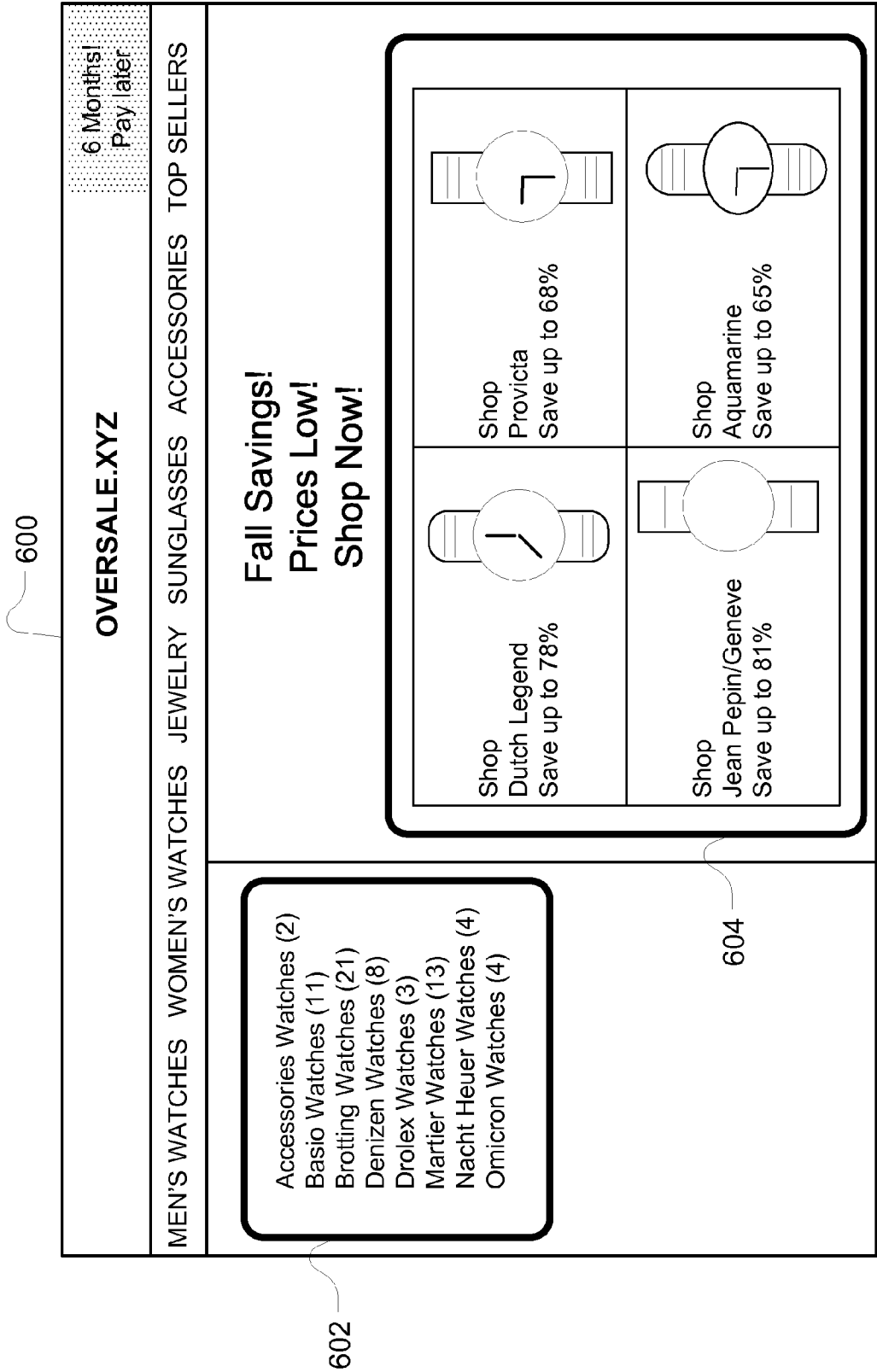


FIG. 6B



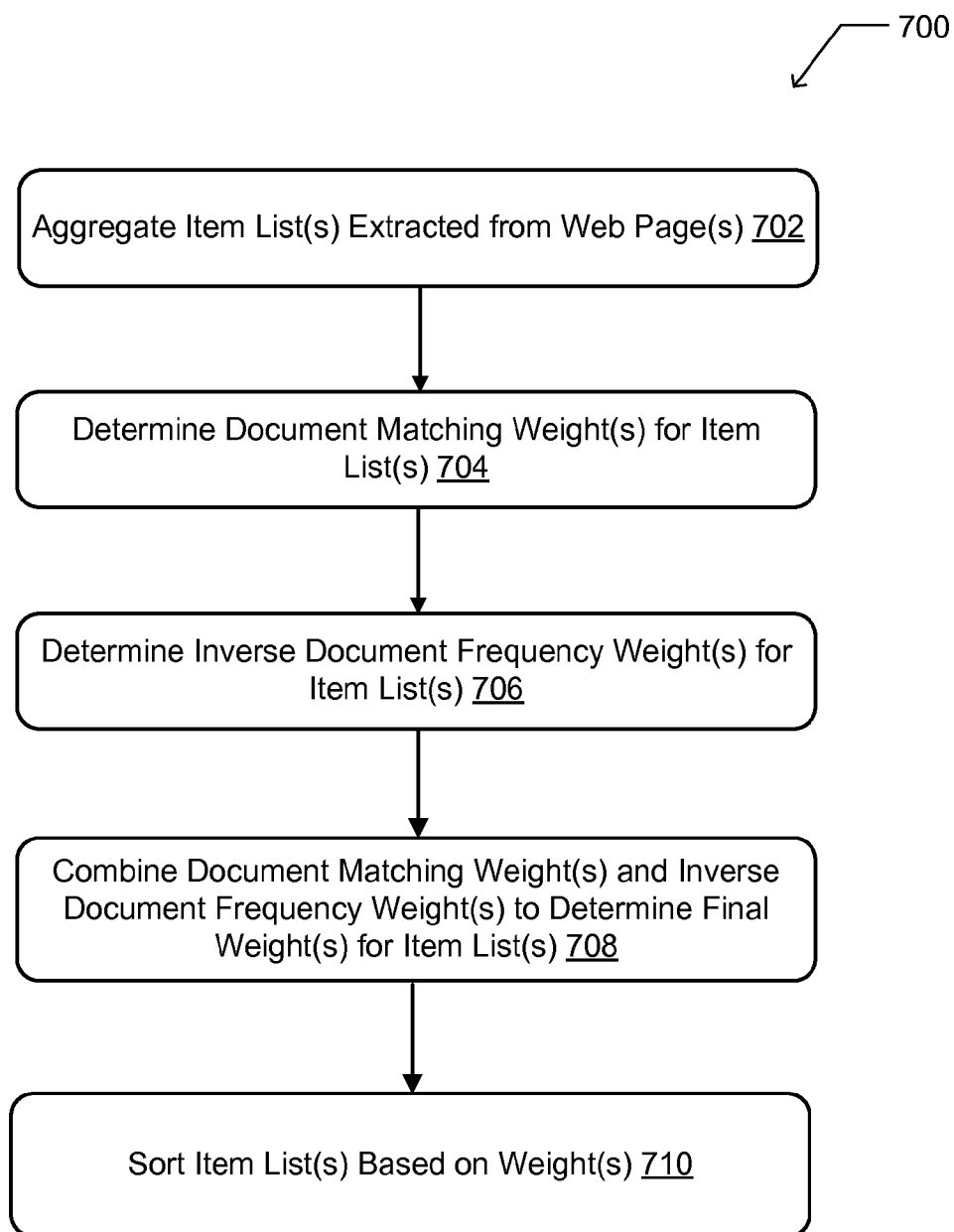


FIG. 7

## EXTRACTING QUERY DIMENSIONS FROM SEARCH RESULTS

### BACKGROUND

**[0001]** Web search engines provide useful tools to enable users to find information on the web. Traditional search services enable a user to enter a search query in the form of one or more search terms and zero or more logical operators. After entering the query, the user may send a command to the search engine to execute a search based on the query. The search engine may then search the web for documents that satisfy the query to some degree, and provide the list of result documents to the user in the form of a list of identifiers such as Uniform Resource Locators (URLs), Uniform Resource Identifiers (URIs), Internet Protocol (IP) addresses, or other identifiers. The results list may also include excerpts from the result documents, descriptions, ranking information, advertisements, social networking information, and other information. Unfortunately, in many cases the search results may be noisy or lengthy, making it difficult for the user to find the desired information in the search results.

### SUMMARY

**[0002]** This application describes techniques for extracting or mining query dimensions from search results. As used herein, a query dimension is a set of items (e.g., words, terms and/or phrases) that describe and/or summarize an aspect of a query. One or more query dimensions may be extracted from a set of web pages resulting from the execution of a search query, and such query dimensions may provide useful information about the query from various perspectives. The extracted query dimensions may then be stored and/or provided to a user to help the user more readily understand the query results and various aspects of the query, and to guide subsequent searches.

**[0003]** Extracting or mining query dimensions may include extracting item lists from the set of search results documents (e.g., web pages or web sites). This extraction may be based on an analysis of the documents to identify free text patterns (e.g., regular expressions), metadata tag patterns, and/or repeated region patterns within the documents. The extracted item lists may then be weighted to determine which lists are more or less important. For example, the weighting for a particular list may be based on a frequency of occurrence of the list's items in the results page, and/or occurrence of the list's items in a results page that appears higher in the search results. Weighting methods may include a document matching weight, an average invert document frequency weight, or a combination of the two. These methods are described in more detail below. The weighted lists may be clustered based on similar or identical items included in multiple lists to generate query dimensions, and the items within each query dimension may be ordered or ranked. The query dimensions may also be ranked based on their quality, and a predetermined number of the top-ranked query dimensions may be stored or provided to the user.

**[0004]** The determination of query dimensions may be performed in an online mode in response to a real-time user query. Query dimension extraction may also be performed in an offline mode for common queries, and the determined query dimensions may be stored and provided to future users requesting a search query.

**[0005]** This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

### BRIEF DESCRIPTION OF THE DRAWINGS

**[0006]** The detailed description is described with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same reference numbers in different figures indicate similar or identical items.

**[0007]** FIG. 1 illustrates an example search engine user interface, with example query dimensions generated based on the search results.

**[0008]** FIG. 2 is a schematic diagram depicting an example environment in which embodiments may operate.

**[0009]** FIG. 3 is a diagram of an example computing device, in accordance with embodiments, that may be deployed as part of the example environment of FIG. 2.

**[0010]** FIG. 4 depicts a flow diagram of an illustrative process for generating query dimensions, in accordance with embodiments.

**[0011]** FIG. 5 depicts a flow diagram of an illustrative process for extracting item lists from web pages, in accordance with embodiments.

**[0012]** FIGS. 6A and 6B depict an example web site from which item lists may be extracted, in accordance with embodiments.

**[0013]** FIG. 7 depicts a flow diagram of an illustrative process for weighting item lists, in accordance with embodiments.

### DETAILED DESCRIPTION

#### Overview

**[0014]** Embodiments are directed to the extraction of query dimensions from search results, to provide an improved search experience for users. Because each extracted query dimension includes a set of items that summarize an aspect of a search query, displaying query dimensions alongside search results may enable a user to understand important aspects of a search query without the need to browse multiple search result pages.

**[0015]** FIG. 1 illustrates example query result web pages and query dimensions extracted from and displayed with the result pages. User interface 100 is an example user interface for a search engine. The user interface 100 includes a query entry field 102, where a user has entered the search query "watches". The search engine has identified a list of result documents (e.g., web pages), and displayed them in results 104. As shown, various query dimensions have been extracted from the results. Dimension 106 is a list of items related to brands of watches. Dimension 108 is a list of items related to gender styles of watches. Dimension 110 is a list of items related to watch functionality types. Dimension 112 is a list of items related to formality and/or usage styles of watches. Dimension 114 is a list of items related to watch colors. In this example, the various extracted query dimensions each include a list of items that illustrate an aspect of the search query "watches". Displaying the query dimensions alongside the search results provides a user with a way of learning about

different brands, types, styles, and/or categories of watches without the need to browse multiple web pages.

**[0016]** Moreover, in some embodiments the query dimensions themselves may be displayed as hyperlinks to facilitate further (e.g., more narrow) searches based on the query dimensions. For example, as shown in FIG. 1, query dimension **106** includes “martier” brand watches as an item, and a user may click on a hyperlink “martier” to request a subsequent search on “watches & martier”. Though not shown in FIG. 1, in some embodiments user interface **100** may include features to enable a user to select multiple query dimensions for a subsequent search and/or to combine multiple query dimensions using various logical operators to compose a search query.

**[0017]** In some embodiments, the displayed query dimensions may provide a direct answer to a user question included in a query. For example, a user may query on “TV show season 5” and a generated query dimension may include items that are episode titles from season five of the television show. In such instances, displaying query dimensions may save the user additional browsing and/or searching, given that the query dimension itself directly provides the desired information. Extraction of query dimensions from search result web pages is described in more detail below, with reference to FIGS. 2-7.

#### Illustrative Environment

**[0018]** FIG. 2 shows an example environment **200** in which embodiments may operate. As shown, the various devices of environment **200** communicate with one another via one or more networks **202** that may include any type of networks that enable such communication. For example, networks **202** include public networks such as the Internet, private networks such as an institutional and/or personal intranet, or some combination of private and public networks. Networks **202** also include any type of wired and/or wireless network, including but not limited to local area networks (LANs), wide area networks (WANs), Wi-Fi, WiMax, and mobile communications networks (e.g. 3G, 4G, and so forth). Networks **202** may utilize communications protocols, including packet-based and/or datagram-based protocols such as internet protocol (IP), transmission control protocol (TCP), user datagram protocol (UDP), or other types of protocols. Moreover, networks **202** may also include a number of devices that facilitate network communications and/or form a hardware basis for the networks, such as switches, routers, gateways, access points, firewalls, base stations, repeaters, backbone devices, and the like.

**[0019]** Environment **200** further includes one or more client device(s) **204** associated with end user(s). Client device(s) **204** include any type of computing device that a user may employ to send and receive information over networks **202**. For example, client device(s) **204** include, but are not limited to, desktop computers, laptop computers, tablet computers, wearable computers, media players, automotive computers, mobile computing devices, smart phones, personal data assistants (PDAs), game consoles, mobile gaming devices, set-top boxes, and the like.

**[0020]** Client device(s) **204** generally include one or more applications, including but not limited to word processing applications, games, web browsers, e-mail client applications, text messaging applications, chat or instant messaging (IM) clients, and other applications. One or more of these applications may include search functionality as part of a user

interface, to enable the user to input a search query, request that a search be performed based on the search query, and display search results and/or query dimensions.

**[0021]** As shown, environment **200** may further include one or more web server device(s) **206**. Web server device(s) **206** include computing devices that are configured to serve content and/or provide services to users over network(s) **202**. Such content and services include, but are not limited to, hosted static and/or dynamic web pages, social network services, e-mail services, chat services, blogging services, games, multimedia, and any other type of content, service or information that may be provided over networks **202**.

**[0022]** Environment **200** also includes one or more search server device(s) **208**. Search server device(s) **208** may be configured (e.g., with a search engine) to receive and execute web search queries entered by users and provide search results. In some embodiments, search server device(s) **208** perform query dimension extraction, generation, and/or mining as described further herein. In other embodiments, query dimension extraction is performed by one or more devices that are separate from search server device(s) **208**, such as dimension extraction server device(s) **210**. Dimension extraction server device(s) **210**, as well as the other types of devices shown in FIG. 2, are described in greater detail herein with regard to FIG. 3.

**[0023]** In some embodiments, environment **200** also includes one or more databases or other data storage device(s) **212**, configured to store data related to the various operations described herein. Such storage devices may be incorporated into one or more of the other devices depicted, or may be external storage devices separate from but in communication with one or more of the devices. For example, data storage device(s) **212** may store search query data and/or query dimension data generated by search server device(s) **208** and/or dimension extraction server device(s) **210**.

**[0024]** Each of the one or more of the devices depicted in FIG. 2 may include multiple computing devices arranged in a cluster, server farm, cloud, or other grouping to share workload. Such groups of devices may be load balanced or otherwise managed to provide more efficient operations. Moreover, although various computing devices of environment **200** are described as clients or servers, each device may operate in either capacity to perform operations related to various embodiments. Thus, the description of a device as client or server is provided for illustrative purposes, and does not limit the scope of activities that may be performed by any particular device. Moreover, in some embodiments one or more of the devices of environment **200** may be combined. For example, search server device(s) **208** may be combined with dimension extraction server device(s) **210**.

#### Illustrative Computing Device Architecture

**[0025]** FIG. 3 depicts a diagram for an example computer system architecture for one or more of the devices depicted in FIG. 2. As shown, computing device **300** includes processing unit **302**. Processing unit **302** may encompass multiple processing units, and may be implemented as hardware, software, or some combination thereof. Processing unit **302** may include one or more processors. As used herein, processor refers to a hardware component. Processing unit **302** may include computer-executable, processor-executable, and/or machine-executable instructions written in any suitable programming language to perform various functions described

herein. In some embodiments, processing unit **302** may further include one or more graphics processing units (GPUs).

**[0026]** Computing device **300** further includes a system memory **304**, which may include volatile memory such as random access memory (RAM), static random access memory (SRAM), dynamic random access memory (DRAM), and the like. System memory **304** may further include non-volatile memory such as read only memory (ROM), flash memory, and the like. System memory **304** may also include cache memory. As shown, system memory **304** includes one or more operating systems **306**, and one or more executable components **310**, including components, programs, applications, and/or processes, that are loadable and executable by processing unit **302**. System memory **304** may further store program/component data **308** that is generated and/or employed by executable components **310** and/or operating system **306** during their execution.

**[0027]** Executable components **310** include one or more components to implement functionality described herein on one or more of the devices depicted in FIG. 2. For example, executable components **310** may include a search engine component **312** that receives a search query from a user, executes the search query, and provides a list of search result documents (e.g., web pages) to the user. In some embodiments, search engine component **312** includes a user interface to enable the user to enter a search query and view the search results. In some embodiments, search engine component **312** also provides a list of query dimensions that have been extracted from result pages, as described below.

**[0028]** In some embodiments, executable components **310** also include an extraction component **314** that operates to extract one or more item lists from web pages, as part of a process to determine query dimensions. Executable components **310** may also include a weighting component **316** to weight the extracted item lists, a clustering component **318** to cluster or otherwise combine item lists based on similar items within the lists to generate query dimensions, and a ranking component **320** to rank query dimensions and/or rank items within the query dimensions based on their frequency and/or importance. The operation of each of these components is described in greater detail below.

**[0029]** Executable components **310** may further include other components **322**. In various embodiments, executable components **310** may be distributed to operate on one device or on more than one device, in virtually any combination. Thus, the depiction of executable components **310** on the single computing device **300** in FIG. 3 should not be construed as limiting.

**[0030]** As shown in FIG. 3, computing device **300** may also include removable storage **330** and/or non-removable storage **332**, including but not limited to magnetic disk storage, optical disk storage, tape storage, and the like. Disk drives and associated computer-readable media may provide non-volatile storage of computer readable instructions, data structures, program modules, and other data for operation of computing device **300**.

**[0031]** In general, computer-readable media includes computer storage media and communications media.

**[0032]** Computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structure, program modules, and other data. Computer storage media includes, but is not limited to, RAM, ROM, erasable programmable

read-only memory (EEPROM), SRAM, DRAM, flash memory or other memory technology, compact disc read-only memory (CD-ROM), digital versatile disks (DVDs) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other non-transmission medium that can be used to store information for access by a computing device.

**[0033]** In contrast, communication media may embody computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transmission mechanism. As defined herein, computer storage media does not include communication media.

**[0034]** Computing device **300** may include input device(s) **334**, including but not limited to a keyboard, a mouse, a pen, a game controller, a voice input device for speech recognition, a touch input device, and the like. Computing device **300** may further include output device(s) **336** including but not limited to a display, a printer, audio speakers, a haptic output, and the like. Computing device **300** may further include communications connection(s) **338** that allow computing device **300** to communicate with other computing devices **340**, including client devices, server devices, databases, and/or other networked devices available over network(s) **202**.

#### Illustrative Processes

**[0035]** FIGS. 4, 5, and 7 depict flowcharts showing example processes in accordance with various embodiments. The operations of these processes are illustrated in individual blocks and summarized with reference to those blocks. The processes are illustrated as logical flow graphs, each operation of which may represent one or more operations that can be implemented in hardware, software, or a combination thereof. In the context of software, the operations represent computer-executable instructions stored on one or more computer storage media that, when executed by one or more processors, enable the one or more processors to perform the recited operations. Generally, computer-executable instructions include routines, programs, objects, modules, components, data structures, and the like that perform particular functions or implement particular abstract data types. The order in which the operations are described is not intended to be construed as a limitation, and any number of the described operations can be combined in any order, subdivided into multiple sub-operations, and/or executed in parallel to implement the described processes.

**[0036]** FIG. 4 depicts an example process **400** for extracting query dimensions from one or more web pages for a search query. In some embodiments, process **400** may be performed by one or more components executing on search server device(s) **208** or dimension extraction server device(s) **210**, such as extraction component **314**, weighting component **316**, clustering component **318**, and/or ranking component **320**.

**[0037]** At **402** one or more document(s) are identified resulting from a search performed based on a search query. The results document(s) may include web page(s) or other forms of content, and may be listed with identifying information such as a URL, URI, IP address, and the like. In some embodiments, the search results are not limited to a particular domain such as products, people, and the like. Such an open domain approach may produce a more comprehensive collection of query dimensions for the search query. In some embodiments, a certain number (e.g., five) of the top result

pages are selected for query dimension extraction. In some embodiments, the result documents may include one or more web sites that each include multiple web pages.

**[0038]** At **404** one or more item list(s) are automatically extracted from the result web page(s) identified at **402**. An item list is a list of words, terms, or phrases. In many cases, information in a web page or other document is provided in the form of a list to increase usability and aesthetic quality of the document. Certain patterns may be used to identify lists in web pages. This pattern-based extraction of item list(s) is described in greater detail with regard to FIGS. 5, 6A, and 6B.

**[0039]** FIG. 5 depicts an example process **500** for extracting item list(s) from documents. In some embodiments, item list extraction is based on recognition of free text patterns, metadata tag patterns, and/or repeated region patterns within a document. At **502** one or more item lists are extracted from the web page(s) based on free text patterns identified within the web page(s). In some embodiments, this includes the extraction of all text from a document and a splitting of the text into sentences based on periods or other sentence delimiters.

**[0040]** One or more patterns (e.g., regular expressions) may then be employed to extract matching list items from each sentence. In some embodiments, a search may be performed for a pattern “item {, item}\* (and/or) {other} item” in each sentence of a document. For example, if a web page includes the sentence “We shop for gorgeous wrist watches from Meiko, Mulova, Brotting, Denizen, Drolex, or Provieta.” This may lead to an item list consisting of “Meiko, Mulova, Brotting, Denizen, Drolex, Provieta.”

**[0041]** In some embodiments, a pattern “{^item (:|-).+\$)+” is employed to extract lists from semi-structured paragraphs. Such a pattern may be used to extract lists from continuous lines that are comprised of two parts separated by a dash or a colon. In such cases, the first parts of these lines may be extracted to form an item list. For example, the following paragraph:

**[0042]** ... are highly important for the following reasons:

**[0043]** Consistency—every fact table is filtered consistently

**[0044]** Integration—queries are able to drill different processes

**[0045]** Reduced development time to market—the common dimensions are available without additional analysis

may be analyzed to extract the list “Consistency, Integration, Reduced development time to market.”

**[0046]** At **504** one or more item lists are extracted from the web page(s) based on metadata tag patterns identified within the web page(s). The web page(s) may include metadata according to various markup languages such as Hypertext Markup Language (HTML), Extensible Markup Language (XML), Extensible Hypertext Markup Language (XHTML), LaTeX, GenCode, Generalized Markup Language (GML), Standard Generalized Markup Language (SGML), Scribe, or other forms of metadata. In some embodiments, item list(s) are extracted from web pages based on an identification of list-style metadata tags in the web pages. For example, HTML tags SELECT, UL, OL, and TABLE may be used to identify and extract item lists from web pages. Table 1 lists example HTML source from which item lists may be extracted.

TABLE 1

---

```

SELECT:
<select name="ProductFinder2" id="ProductFinder2">
<option value="WatchBrands.htm">Watch Brands</option>
<option value="Brands-Basio.htm">Basio</option>
<option value="Brands-Brotting.htm">Brotting</option>
<option value="Brands-Denizen.htm">Denizen</option>
<option value="Brands-Drolex.htm">Drolex</option>
<option value="Brands-Martier.htm">Martier</option>
UL:
<ul><li><a href="/rst.asp?q=dive">Dive</a></li>
<li><a href="/rst.asp?q=titanium">Titanium</a></li>
<li><a href="/rst.asp?q=automatic">Automatic</a></li>
<li><a href="/rst.asp?q=quartz">Quartz</a></li>
<li><a href="/rst.asp?q=gold">Gold</a></li></ul>
TABLE:
<table width="100%">
<tr><td width="10%"></td><td>White</td></tr>
<tr><td></td><td height="20">Red</td></tr>
<tr><td></td><td height="20">Black</td></tr>
<tr><td></td><td height="20">Pink</td></tr>
<tr><td height="4" colspan="2"></td></tr></table>

```

---

**[0047]** For the SELECT tag, text from child tags (e.g., OPTION tags) may be extracted to form an item list. In some embodiments, the first item is left out of the list if it starts with certain predefined text such as “select” or “choose.” In the example SELECT tag shown in Table 1, the extracted list is “Watch Brands, Basio, Brotting, Denizen, Drolex, Martier.” For UL and OL tags, text is extracted from child tags (e.g., LI tags) to form a list. In the example UL tag of Table 1, the extracted list is “Dive, Titanium, Automatic, Quartz, Gold.”

**[0048]** For the TABLE tag, a list may be extracted from each column and/or each row of the table. Thus, for a table containing m rows and n columns, at most m+n lists may be extracted. In some embodiments, for each column cells within THEAD or TFOOT tags are regarded as table headers and are left out of the extracted list. In some embodiments, the first cell of each column may be left out when its cascading style is different from other cells in the column. In the example TABLE tag of Table 1, the extracted list is “White, Red, Black, Pink.” Although the examples given above specifically describe HTML tags, embodiments are not so limited and other types of metadata tags may be employed to extract item lists.

**[0049]** At **506** one or more item lists are extracted from the web pages based on repeated region patterns within the web pages. In some cases, web page designers organize information on a page in the form of organized blocks or other structures (e.g., visual structures) in a repeated style. In such instances, item lists may be extracted from the blocks based on a determination of a similarity in style (e.g., color, font, line weight, block size, position, and the like) between the blocks. In some embodiments, such similarities are identified based on an examination of the Document Object Model (DOM) tree for a page. Repeat regions (e.g., regions that contain more than one block) are identified based on similarities detected in the DOM tree of a page, and/or in individual DOM trees corresponding to the blocks. Then, the HTML nodes may be extracted from each block and grouped based on tag names and/or display styles, and text from the nodes may be extracted as one or more item lists.

**[0050]** At **508** one or more post-processing steps are performed on the extracted item list(s). In some embodiments, such post-processing includes a normalization of the items in the lists by removing unneeded symbol characters (e.g., “[” and “]”), converting uppercase letters to lowercase, and

removal of long items that contain more than a certain number (e.g., 20) terms. In some embodiments, post-processing includes removal of lists that contain less than a certain number (e.g., 2) of unique items, and/or removal of lists that contain more than a certain number (e.g., 200) of unique items.

**[0051]** Embodiments may employ free text, metadata tag, and repeated region pattern recognition techniques individually or in any combination, and may further employ other techniques for item list extraction. FIGS. 6A and 6B illustrate an example web page and the extraction of item lists from the web page. FIG. 6A depicts an example web page 600 associated with online merchant “oversale.xyz” selling watches and other items. FIG. 6B shows the same web page 600 with a graphical illustration of item list extraction. Two item lists are shown extracted from web page 600. Item list 602 is an example of a list extracted through the metadata tag pattern recognition described above. Item list 604 is an example of a list extracted through the repeated region pattern recognition described above.

**[0052]** Returning to FIG. 4, at 406 the one or more extracted item lists are weighted. In some embodiments, such weighting is based on a frequency of occurrence of the item lists or items of the item lists in the web pages. Moreover, in some embodiments the weighting is based on which pages the item lists were extracted from. For example, item lists extracted from higher ranked search result pages may be given a greater weight than item lists extracted from lower ranked pages. Weighting may also serve as a means to remove less informative (e.g., noise) lists from the analysis. Table 2 provides examples of less informative lists that may be removed through weighting.

TABLE 2

List ID	List items (separated by commas)
1	we recommend, my account, help
2	home, customer service, my account, tracking, FAQs
3	read, edit, view, history
4	brovado 605635 luno two tone . . . 547.50 717.00 1 rating 1 review, brovado museum strap 0690299 . . . 225.00 395.00 1 rating, denizen caliber 2100 av0031 . . . 299.00 350.99 11 ratings

**[0053]** In this example, the first three lists of Table 2 were derived from navigational links designed to help users navigate between pages of a web site. The fourth list is an example of a list extraction error in which several types of information are included in the same item list. Some embodiments, remove such lists from the analysis as not informative to the query, by means of the weighting described below. Thus, in some embodiments weighting serves to identify good lists, e.g. those higher quality lists that are at least partially present in multiple web pages and/or contain items that are informative to the query. Such item lists tend to generate more useful query dimensions.

**[0054]** Item list weighting is described in greater detail with regard to FIG. 7, which depicts an example process 700 for item list weighting. At 702 the one or more item lists extracted from the web page(s) are collected in preparation for the weighting analysis. At 704 a document matching weight is determined for each item list. In some embodiments, document matching weight determination is a weighting technique in which a higher weight is assigned to a list based on its

appearance in a more highly ranked search result page. Document matching weight  $S_{DOC}$  may be calculated by Equation 1 below:

$$S_{DOC} = \sum_{d \in R} (s_d^m * s_d^r) \quad (\text{Equation 1})$$

**[0055]** In this equation,  $d$  is a document (e.g., web page) in the set of results  $R$ .  $s_d^m$  is the percentage of items of the list contained in document  $d$ . In some embodiments, a list  $l$  is determined to be supported by document  $d$  if  $d$  contains all items of list  $l$ , or more than a certain number of items in the list. The more items of the list  $d$  contains, the stronger  $d$  supports the list. If  $|d \cap l|$  is the number of shared items in  $d$  and  $l$ , and  $|l|$  is the number of items contained in list  $l$ , then  $s_d^m$  may be given by Equation 2:

$$s_d^m = \frac{|d \cap l|}{|l|} \quad (\text{Equation 2})$$

**[0056]** Another term in Equation 1,  $s_d^r$ , measures the importance of document  $d$ , based on the rank of  $d$  within search results  $R$ . Documents ranked higher in the search results tend to be more relevant to the search query, and may be given greater importance by embodiments. In some embodiments,  $s_d^r$  may be given by Equation 3:

$$s_d^r = 1/\sqrt{\text{rank}_d} \quad (\text{Equation 3})$$

In this equation,  $\text{rank}_d$  is the rank of document  $d$ . Thus, the higher  $d$  is ranked in the results list, the larger is score  $s_d^r$ .

**[0057]** At 706 an average inverse document frequency weight is determined for each of the item lists. In some embodiments, this weight reflects the idea that a list has less value if it is composed of common items in the corpus of words in the relevant language. The inverse document frequency weight  $S_{IDF}$  may be calculated according to Equations 4 and 5 below:

$$S_{IDF} = \frac{1}{|l|} \cdot \sum_{e \in l} idf_e \quad (\text{Equation 4})$$

where

$$idf_e = \log \frac{N - N_e + 0.5}{N_e + 0.5} \quad (\text{Equation 5})$$

In these equations,  $N_e$  is the total number of documents that contain the item  $e$  in the corpus, and  $N$  is the total number of documents. In some embodiments, the corpus used is a collection of web pages from archived sites on the web. Various corpora (e.g., the ClueWeb09 collection) may be employed by embodiments.

**[0058]** In some embodiments, at 708 the document matching weight and inverse document frequency weight for each item list may be combined to determine a final weight for each item list, such that the final weight of a list  $l$  is indicated by Equation 6:

$$S_l = S_{DOC} * S_{IDF} \quad (\text{Equation 6})$$

In some embodiments, either the document matching weight or the inverse document frequency weight is used as a final weight instead of using a combination of the two. At 710 the item lists are sorted based on their final weights.

**[0059]** Returning to FIG. 4, at 408 the weighted item lists are clustered to generate one or more query dimensions. In

some embodiments, clustering includes integrating item lists that have similar and/or identical items, such that two item lists are clustered together if they share a sufficient number of items. Some embodiments employ a quality threshold algorithm for clustering. In such cases, a distance  $d_l$  may be defined between two lists  $l_1$  and  $l_2$ , as shown in Equation 7:

$$d_l(l_1, l_2) = 1 - \frac{|l_1 \cap l_2|}{\min\{|l_1|, |l_2|\}} \quad (\text{Equation 7})$$

where  $|l_1 \cap l_2|$  is the number of shared items within  $l_1$  and  $l_2$ . Two item lists may be placed in the same cluster if the distance between them is below a certain threshold. Moreover, in some embodiments a linkage distance between clusters is calculated to determine whether two clusters can be combined into a single cluster. The linkage distance  $d_c$  between two clusters  $c_1$  and  $c_2$  may be calculated using a distance function such as that given by Equation 8:

$$d_c(c_1, c_2) = \max_{l_1 \in c_1, l_2 \in c_2} d_l(l_1, l_2) \quad (\text{Equation 8})$$

In some embodiments, two clusters may be merged if the linkage distance between them is below a certain threshold. Thus, in some embodiments two groups of lists may be merged when every pair of lists between the two groups is similar enough.

**[0060]** The quality threshold algorithm described above assumes that all data is equally important, such that the cluster that has the highest number of data points (e.g., item lists) is selected in each iteration. However, some embodiments employ a modified quality threshold algorithm which diverges from the quality threshold algorithm by assuming that item lists are not equally important. Thus, in some embodiments a modified quality threshold clustering algorithm is used to group similar lists into clusters. Such an algorithm may ensure higher quality clustering by finding large clusters whose diameters do not exceed a particular diameter threshold. In some embodiments, the diameter of a cluster is the longest distance between each pair of data points within the cluster. Such an algorithm may also prevent dissimilar data from being clustered together. In the modified quality threshold algorithm, better lists (e.g., higher weighted lists) are grouped together earlier in the process.

**[0061]** In some embodiments, a maximum diameter  $Dia_{max}$  and a minimum weight  $W_{min}$  are selected for the clusters. Then, a candidate cluster is built for the most important point (e.g., the highest weighted item list) by iteratively including each other point that is closest to group until the diameter of the cluster surpasses the threshold  $Dia_{max}$ . This candidate cluster may then be saved if the total weight of its points  $w_c$  is not smaller than  $W_{min}$ . All points in the cluster may then be removed from further consideration, and the process may repeat recursively with the reduced set of points.

**[0062]** In this way, the modified quality threshold algorithm may operate to identify a larger number of neighbors for the more important points (e.g., item lists that are higher weighted), and therefore clusters may be biased toward the more important points. As an example, suppose six item lists, listed in order of descending weight such that  $S_1 > S_2 > S_3$  and so on:

**[0063]**  $l_1 = (\text{martier, brotting, omicron, denizen})$

**[0064]**  $l^2 = (\text{brotting, omicron, denizen, nacht heuer})$

**[0065]**  $l_3 = (\text{brotting, omicron, denizen, movie, music, book})$

**[0066]**  $l_4 = (\text{movie, music, book})$

**[0067]**  $l_5 = (\text{music, book, radio})$

**[0068]**  $l_6 = (\text{movie, book, radio})$

In this case, the unmodified quality threshold algorithm may ignore the weights of these lists and generate a cluster  $(l_3, l_4, l_5, l_6)$  with  $Dia_{max} = 0.6$ . However, the modified quality threshold algorithm may generate a cluster  $(l_1, l_2, l_3)$  based around the highest weighted point, i.e. list  $l_1$ . This second result may be favorable, particularly in cases where  $S_1$  is much greater than  $S_3$ . Moreover, in some embodiments the modified algorithm may proceed more efficiently than the unmodified algorithm, given that the modified algorithm is more likely to generate a single candidate cluster whereas the unmodified algorithm generates a candidate cluster for each remaining item list.

**[0069]** In some embodiments, the weight of a cluster is computed based on a number of web pages from which its lists are extracted. Further, in some embodiments web sites (e.g., of multiple web pages) may be considered in the clustering process instead of individual web pages, because web pages from the same web site may share similar or identical page templates and thus contribute duplicate lists to the process. In some embodiments,  $Dia_{max} = 0.6$  and  $W_{min} = 3$ , such that lists of a qualified cluster are from at least three unique web sites. However, other values may be employed by embodiments. In some embodiments, after clustering is complete the clustered lists are identified as candidate query dimensions.

**[0070]** At **410** the candidate query dimensions are ranked. In some embodiments, query dimension ranking is based on two criteria. First, a dimension may be ranked higher if the item lists that formed the dimension were extracted from more unique web sites or web pages. Second, a dimension may be ranked higher if the item lists that formed the dimension are more important (i.e., have higher weights). Based on these criteria, an importance  $S_d$  of dimension  $d$  may be given by Equation 9:

$$S_d = \sum_{l \in \text{Sites}(d)} \max_{l \in S_l} S_l \quad (\text{Equation 9})$$

where  $S_l$  is the weight of a list  $l$ , and  $\text{Sites}(d)$  is the collection of all the web sites (or web pages) that were considered in generating dimension  $d$ .

**[0071]** At **412** the one or more items within a query dimension are ranked. In some embodiments, the importance of a particular item within a query dimension depends on how many item lists contained the item and where the item ranked in those lists, given that a particular item may have been placed higher in a list by the web site or web page designer if that item is more important in some way. In some embodiments, the weight  $S_{e|d}$  of an item  $e$  within a dimension  $d$  is given by Equation 10:

$$S_{e|d} = \sum_{s \in \text{Sites}(d)} w(d, e, s) = \sum_{s \in \text{Sites}(d)} \frac{1}{\sqrt{\text{AvgRank}_{d,e,s}}} \quad (\text{Equation 10})$$

where  $w(d, e, s)$  is the weight contributed by web site (or web page)  $s$ , and  $\text{AvgRank}_{d,e,s}$  is the average rank of item  $e$  within all lists extracted from web site (or web page)  $s$ .

**[0072]** In some embodiments, the items in each query dimension are sorted based on their determined weights. In some embodiments, there is a further step of determining one or more qualified items of a dimension based on whether the

weight for an item is above a certain threshold. Moreover, in some embodiments, a qualified item may also be dependent on whether the item occurred first in at least one list from at least one web page, and/or whether the item is present in a list from at least one other web page.

[0073] At 414 in some embodiments the determined candidate query dimensions may be further filtered to determine one or more high-quality query dimensions. In some embodiments this determination may be based on whether a query dimension has a weight above a certain quality threshold, the weight determined at 410. Further, in some embodiments determination of high-quality query dimensions may be based on the particular application that the user is running to request the search. Once the high-quality query dimensions are determined, they may be stored in a database and/or provided for display to the user alongside the top search results for the user's search query.

[0074] Moreover, in some embodiments a predetermined number of the highest ranked dimensions (e.g., those with the highest weight) may be stored and/or provided to the user, and this predetermined number may be based on the particular application that the user is running to request the search. In those embodiments where the high-quality dimensions are determined as those with a weight above a predetermined quality threshold, the quality threshold may be based on the particular application.

[0075] In some embodiments, extraction of query dimensions as described above may occur online and/or in real-time, in response to a user query. However, such online processing may be resource intensive and may increase the time required to respond to the user's query. Given that, in some embodiments, extraction of query dimensions may be performed in an offline mode for common search queries, and the results may be stored in a database to be provided to users at a future time.

## CONCLUSION

[0076] Although the techniques have been described in language specific to structural features and/or methodological acts, it is to be understood that the appended claims are not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as example forms of implementing such techniques.

What is claimed is:

1. A computer-implemented method comprising:
  - identifying a plurality of web pages resulting from a search based on a search query;
  - employing at least one processor to automatically extract one or more item lists from the plurality of web pages;
  - weighting the one or more item lists;
  - clustering the one or more item lists to generate one or more query dimensions that each summarize an aspect of the search query, the clustering based on a determination that at least some of the item lists include one or more similar items; and
  - ranking one or more items within each of the one or more query dimensions.
2. The method of claim 1, further comprising:
  - weighting the one or more query dimensions; and
  - identifying one or more high-quality query dimensions as the one or more query dimensions that have a weight higher than a quality threshold value.

3. The method of claim 1, wherein extracting the one or more item lists is based on identifying one or more free text patterns within the plurality of web pages.

4. The method of claim 1, wherein extracting the one or more item lists is based on identifying one or more metadata tag patterns within the plurality of web pages.

5. The method of claim 1, wherein extracting the one or more item lists is based on identifying one or more repeated region patterns within the plurality of web pages.

6. The method of claim 1, wherein weighting the one or more item lists is based on a document matching weight.

7. The method of claim 1, wherein weighting the one or more item lists is based on an inverse document frequency weight.

8. The method of claim 1, wherein weighting the one or more item lists is based on a combination of a document matching weight and an inverse document frequency weight.

9. The method of claim 1, wherein ranking the one or more items within each of the one or more query dimensions is based on a frequency of the one or more items within the one or more item lists.

10. A system comprising:

at least one processor;

an extraction component that executes on the at least one processor to extract one or more item lists from a plurality of web pages resulting from execution of a search query;

a weighting component that executes on the at least one processor to weight each of the one or more item lists; and

a clustering component that executes on the at least one processor to generate one or more query dimensions by clustering the one or more item lists based on a determination that at least some of the item lists include one or more similar items.

11. The system of claim 10, further comprising a search engine component that executes on the at least one processor to receive the search query and execute a search based on the search query.

12. The system of claim 10, further comprising a ranking component that executes on the at least one processor to rank one or more items within each of the one or more query dimensions based on a frequency of the one or more items within the one or more item lists.

13. The system of claim 10, further comprising a ranking component that executes on the at least one processor to rank the one or more query dimensions.

14. The system of claim 13, wherein the ranking component further operates to identify one or more high-quality query dimensions as a predetermined number of highest ranked query dimensions.

15. The system of claim 10, wherein extracting the one or more item lists is based on identifying within the plurality of web pages at least one of:

one or more free text patterns;

one or more metadata tag patterns; and

one or more repeated region patterns.

16. A computer-implemented method comprising:

employing at least one processor to extract one or more item lists from a plurality of web pages;

weighting each of the one or more item lists; and

clustering the one or more item lists to generate one or more dimensions.



**17.** The method of claim **16**, further comprising identifying at least one of the one or more dimensions to be displayed within an application, the identifying based at least in part on the application.

**18.** The method of claim **16**, wherein the clustering is based on a determination that at least some of the item lists include one or more similar items.

**19.** The method of claim **16**, further comprising ranking one or more items within each of the one or more dimensions based on a frequency of the one or more items within the one or more item lists.

**20.** The method of claim **16**, wherein weighting the one or more item lists is further based on at least one of a document matching weight and an inverse document frequency weight.

\* \* \* \* \*