

RETRIEVAL OF ATTRIBUTE VALUES BASED UPON IDENTIFIED ENTITIES**BACKGROUND**

[0001] Computer-executable spreadsheet applications are conventionally configured to allow users thereof to organize and analyze tabular data. Generally, a spreadsheet of a computer-executable spreadsheet application includes numerous cells arranged in rows and columns, wherein each cell can retain a respective value (or function that outputs a value based upon data in other cells). A user of the spreadsheet application may select a cell and manually modify content thereof, either by specifying a new value to be included in the cell or by modifying a function used to compute a value for the cell.

[0002] Conventional spreadsheet applications are well suited for organizing and manipulating tabular data, particularly when the user is familiar with functions supported by the spreadsheet applications (and commands that invoke such functions). In an example, the user can cause values in a respective plurality of cells to be summed by setting forth a textual command, such as “SUM”, and subsequently identifying cells that include values that are desirably included in a summation.

[0003] In conventional spreadsheet application, however, it can be arduous to augment a table with data that is not already loaded into the spreadsheet application. For example, a novice user may load a table into the spreadsheet application, where the table includes a column of entities (e.g., where an entity is a string that is representative of a person, place, or thing). Thus, in an example, the column of entities can include city names, state names, company names, etc. As can be ascertained, a person, place, or thing can be described by an attribute. For instance, the state of California can be described by numerous attributes, such as population (e.g., in a certain year), average rainfall (e.g., over several years), etc., wherein the attributes have values respectively corresponding thereto. Conventionally, for example, to augment a table to add a column of attribute values in correspondence with respective entities, a user must perform a search for the desired attribute values (e.g., through use of a conventional search engine), identify a data source that includes the attribute values (e.g., a table or set of tables), determine that the data source is trustworthy, and then manually transport data from the identified source into the table at the appropriate locations. It can be ascertained that this can be a time-consuming and frustrating task.

SUMMARY

[0004] The following is a brief summary of subject matter that is described in greater detail herein. This summary is not intended to be limiting as to the scope of the claims.

[0005] Described herein are various technologies pertaining to data finding data (DFD)

5 searches, as well as mechanisms for viewing and manipulating data (e.g., tabular data) identified during DFD searches. In an exemplary embodiment, an application executing on a client computing device can include a plurality of machine-readable entities, wherein an entity is a string that is representative of a person, place, or thing. For example, the application can be a spreadsheet application, and the entities may be included in a column
10 of a table loaded into the spreadsheet application. The client computing device may then receive an indication (e.g., from a user or another program executing on the client computing device) that the entities have been selected. Furthermore, the client computing device can (optionally) receive an attribute (e.g., a string representative of an attribute of people, places, or things represented by the entities) specified by the user or program.

15 [0006] The application can then transmit the entities (and the attribute) to a search system. For example, the application can construct a query that includes the entities. In another example, the application can transmit the entities and the attribute to the search system, and the search system can construct the query. The search system executes a search over an index of tables based upon the query. The search system can identify a
20 source table that includes attributes and/or attribute values that may be relevant to the entities used to form the query (and optionally the identified attribute name), and can transmit at least a portion of the source table to the application executing on the client computing device. In an exemplary embodiment, when the search system receives an attribute identified by the user, the search system can identify a source table that includes
25 at least a subset of the specified entities, can extract values of the specified attribute from the source table, and can transmit such values to the application executing on the client computing device. The attribute values can then be presented on a display or stored in a table in a computer-readable storage medium, in appropriate correlation with the entities. Accordingly, the table from which the entities were initially selected can be relatively
30 quickly augmented with values of an attribute identified by the user.

[0007] In another exemplary embodiment, responsive to receiving the entities (and optionally an attribute), the search system can identify a source table that includes values of the attribute for the entities, and can generate a table snippet that summarizes the source table. The table snippet can include, for example, a title of the source table, a source

location of the source table (e.g., a URL of a web page that includes the source table), and a representative portion of the table. The table snippet can be transmitted to the computer-executable application executing on the client computing device, which can display the table snippet on a display for review by a user. The user can review the table snippet to determine whether any content of the source table (summarized by the table snippet) is to be incorporated into the table loaded into the spreadsheet application (e.g., whether a column is to be added to the table loaded into the spreadsheet application). For instance, the table snippet can include a selectable button positioned in graphical relation to a column summarized in the snippet.

[0008] Responsive to the selectable button being selected by the user, attribute values in the column can be imported into the spreadsheet application and appropriately aligned with the entities in the table – resulting in augmenting the table with a new column that includes the above-mentioned attribute values. The table snippet can additionally include a hyperlink or button that, when selected by the user, causes more of the source table to be presented to the user or causes a browser to load a web page that comprises the source table.

[0009] The above summary presents a simplified summary in order to provide a basic understanding of some aspects of the systems and/or methods discussed herein. This summary is not an extensive overview of the systems and/or methods discussed herein. It is not intended to identify key/critical elements or to delineate the scope of such systems and/or methods. Its sole purpose is to present some concepts in a simplified form as a prelude to the more detailed description that is presented later.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Fig. 1 is a functional block diagram of an exemplary system that is configured to perform a data finding data (DFD) search.

[0011] Fig. 2 is a functional block diagram of an exemplary entity augmentor system that is configured to augment identified entities with attribute values corresponding thereto.

[0012] Fig. 3 is a functional block diagram of an exemplary search system that is configured to identify source tables based upon received entities.

[0013] Fig. 4 is an exemplary graphical user interface of a spreadsheet application that illustrates selection of a plurality of entities in a table loaded into a spreadsheet application.

[0014] Fig. 5 is an exemplary graphical user interface of a spreadsheet application that facilitates initiation of a DFD search based upon selected entities.

[0015] Fig. 6 is an exemplary graphical user interface of a spreadsheet application that illustrates augmentation of entities with attribute values.

5 [0016] Fig. 7 is an exemplary graphical user interface of a spreadsheet application that illustrates exemplary data that can be presented responsive to a cell in a spreadsheet being hovered over by a mouse pointer.

[0017] Fig. 8 is an exemplary graphical user interface of a spreadsheet application that illustrates presentment of table snippets on a display, wherein the table snippets

10 summarize respective source tables that include attribute values related to selected entities.

[0018] Fig. 9 illustrates an exemplary table snippet.

[0019] Fig. 10 is an exemplary graphical user interface of a content page, wherein entities in the content page can be selected to initiate performance of a DFD search.

[0020] Fig. 11 is an exemplary graphical user interface of a search engine vertical that
15 facilitates performance of a DFD search.

[0021] Fig. 12 illustrates an exemplary graphical user interface that facilitates performance of a DFD search by way of dragging and dropping graphical objects that are representative of people, places, or things.

[0022] Fig. 13 is a flow diagram that illustrates an exemplary methodology for
20 performing a DFD search.

[0023] Fig. 14 is a flow diagram illustrating an exemplary methodology for augmenting a table that includes a plurality of entities with respective values of an attribute of the entities.

[0024] Fig. 15 is a flow diagram that illustrates an exemplary methodology for
25 presenting a ranked list of attributes on a display, wherein the ranked list of attributes are found to be relevant to a plurality of identified entities.

[0025] Fig. 16 is a flow diagram that illustrates an exemplary methodology for generating table snippets that respectively summarize source tables found to be relevant to selected entities.

30 [0026] Fig. 17 is a flow diagram that illustrates an exemplary methodology for augmenting tabular data based upon selection of an attribute in a ranked list of attributes.

[0027] Fig. 18 is an exemplary computing system.

DETAILED DESCRIPTION

[0028] Various technologies pertaining to data-finding-data (DFD) searches are now described with reference to the drawings, wherein like reference numerals are used to refer to like elements throughout. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of one or more aspects. It may be evident, however, that such aspect(s) may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate describing one or more aspects. Further, it is to be understood that functionality that is described as being carried out by certain system components may be performed by multiple components. Similarly, for instance, a component may be configured to perform functionality that is described as being carried out by multiple components.

[0029] Moreover, the term “or” is intended to mean an inclusive “or” rather than an exclusive “or.” That is, unless specified otherwise, or clear from the context, the phrase “X employs A or B” is intended to mean any of the natural inclusive permutations. That is, the phrase “X employs A or B” is satisfied by any of the following instances: X employs A; X employs B; or X employs both A and B. In addition, the articles “a” and “an” as used in this application and the appended claims should generally be construed to mean “one or more” unless specified otherwise or clear from the context to be directed to a singular form.

[0030] Further, as used herein, the terms “component” and “system” are intended to encompass computer-readable data storage that is configured with computer-executable instructions that cause certain functionality to be performed when executed by a processor. The computer-executable instructions may include a routine, a function, or the like. It is also to be understood that a component or system may be localized on a single device or distributed across several devices. Further, as used herein, the term “exemplary” is intended to mean serving as an illustration or example of something, and is not intended to indicate a preference.

[0031] With reference to Fig. 1, an exemplary system 100 that facilitates performance of a DFD search is illustrated. A DFD refers to a search where a query employed to perform the search is based upon data loaded into an application. Thus, in a DFD search, data loaded into an application is employed to locate related data.

[0032] The system 100 includes a client computing device 102, which can be or include a desktop computing device, a mobile computing device (such as a laptop computing

device, a mobile telephone, a tablet computing device, a wearable computing device, etc.), or other suitable client computing device. The system 100 further includes a server computing device 104 that is in communication with the client computing device 102 by way of a network 106. The network 106 can be the Internet, a cellular network, etc.

5 **[0033]** The client computing device 102 includes a client processor 108 and a client memory 110, wherein the client processor 108 is configured to execute instructions loaded into the client memory 110. The client memory 110 can have an application 112 loaded therein, such that the client processor 108 can execute the application 112. In an exemplary embodiment, the application 112 can be a spreadsheet application. In another
10 example, the application 112 can be a web browser that is configured to load a spreadsheet application. It is to be understood, however, that the application 112 is not limited to being a spreadsheet application or an application (e.g., a browser) that can load a spreadsheet application. For example, the application 112 can be a word processing application, an application that is configured to present content to a user (e.g., a news application), a
15 dedicated e-mail application, an instant messaging application, a video game, a social networking application, etc.

[0034] The application 112 comprises an entity augmentor system 114. Briefly, the application 112 can have a plurality of entities loaded therein, and the entity augmentor system 114 facilitates retrieval of values of an attribute of the entities. With more
20 particularity, an entity is a computer-readable string that is representative of a person, place, or thing. In a non-limiting example, the application 112 can be a spreadsheet application that has a table loaded therein, wherein a column of the table includes entities that identify countries. The entity augmentor system 114 facilitates retrieval of values of an attribute (e.g., population, average temperature, gross domestic product, etc.) of the
25 countries.

[0035] The server computing device 104 includes a server processor 116 and a server memory 118, wherein the server processor 116 executes instructions loaded into the server memory 118. The server memory 118 can include a search system 120 that facilitates performance of a DFD search based upon a plurality of entities (and optionally an
30 attribute) received from the entity augmentor system 114. The server computing device 104 includes or has access to a data store 122 that comprises an index 124, wherein the index 124 is a searchable index of source tables (e.g., wherein the source tables can be two-dimensional tables or multi-dimensional tables). Source tables indexed by the index 124 can be accessible by way of a network. For instance, the source tables can be tables on

web pages that are accessible by way of the Internet. Thus, the search system 120 can receive a plurality of entities (and optionally an attribute) from the entity augmentor system 114, and can execute a search over the index 124 to identify a source table or source tables that are relevant to the entities. For instance, the identified source table or
5 tables can include at least a subset of the received entities and a row or column that includes attribute values that respective correspond to the subset of the received entities.

[0036] It is to be understood that the architecture of the system 100 set forth in Fig. 1 is exemplary. For example, the search system 120 may be a portion of the application 112 loaded into the client memory 110 on the client computing device 102. In another
10 example, the data store 122 that includes the index 124 may be local to the client computing device 102. In still yet another example, the entity augmentor system 114 can be included in the server memory 118. Still further, the entity augmentor system 114, the search system 120, and/or the index 124 can be distributed between the client computing device 102 and the server computing device 104.

[0037] Exemplary operation of the system 100 is now set forth. The application 112, loaded into the client memory 110 and executed by the client processor 108, can have a plurality of entities therein, which for example, can be displayed on a display (not shown) of the client computing device 102. For instance, the application 112 can be spreadsheet application, and the plurality of entities can be included in a column of a spreadsheet of
20 the spreadsheet application. The user, through utilization of an input mechanism (e.g., a mouse, a touch-sensitive display, etc.) for the client computing device 102, can select the plurality of entities, and the application 112 can receive an indication that the plurality of entities have been selected. The entity augmentor system 114 can then receive an indication that a DFD search is to be conducted based upon the plurality of entities. For
25 example, the entity augmentor system 114 can receive the plurality of entities and (optionally) an attribute that is of interest to the user.

[0038] Responsive to receiving the indication, the entity augmentor system 114 can, in an exemplary embodiment, construct a query that can be used to perform a DFD search. The entity augmentor system 114 can then cause the client computing device 102 to
30 transmit the query to the server computing device 104 by way of the network. The query can include, for instance, the entities and the attribute selected by the user. In another example, the entity augmentor system 114 can cause the entities and the attribute (not formed as a query) to be transmitted to the server computing device 104.

[0039] The server 104 receives data from the client computing device 104, which is provided to the search system 120. When the search system 120 receives entities, which are not formatted as a query, the search system 120 can construct a query based upon the entities (and optionally the attribute specified by the user). The search system 120 searches the index 124 based upon the query. The search system 120, based upon the execution of the search over the index 124, identifies at least one source table that is relevant to the query, and is thus relevant to the entities. For example, the at least one source table can include at least a subset of the plurality of entities, known synonyms of entities in the plurality of entities, etc. Further, the at least one source table can include values of an attribute of the plurality of entities, wherein the attribute may have been specified by the user.

[0040] Responsive to identifying the at least one table, the search system 120 causes the server computing device 104 to transmit at least a portion of the source table to the client computing device 102 by way of the network 106. For example, the search system 120 can cause the server computing device 104 to transmit values of the attribute in the identified source table to the client computing device 102. The entity augmentor system 114 receives at least the portion of the source table identified by the search system 120 (e.g., the values of the attribute), and augments the plurality of entities selected by the user with the values of the attribute. For example, when the application 112 is a spreadsheet application and the entities are included in a column of a spreadsheet, the entity augmentor system 114 can add a column to such spreadsheet and populate the column with the values of the attribute, wherein the values of the attribute are appropriately aligned with their respective entities. Other examples of augmenting entities with respective attribute values corresponding thereto will be set forth below.

[0041] Now turning to Fig. 2, a functional block diagram of the entity augmentor system 114 is illustrated. The entity augmentor system 114 includes an input receiver component 202 that receives a selection of a plurality of entities (e.g. from a user or a computer-executable application executing on the client computing device 102). Additionally, optionally, the input receiver component 202 can receive a selection of an attribute, wherein the entity augmentor system 114 is to retrieve values of the attribute for the respective selected entities. The input receiver component 202 can include a transmitter component 204 that transmits the selected entities and the attribute (if provided) to the server computing device 104. As indicated previously, in another exemplary embodiment, the transmitter component 204 can be configured to construct a query based upon the

selected entities and attribute(s), and can be further configured to transmit the query to the server computing device 104.

[0042] The entity augmentor system 114 further includes an output component 206 that, responsive to the input receiver component 202 receiving the selection of the entities, is configured to receive values of an attribute (e.g., the attribute selected by the user) and output the values of the attribute for the selected entities. In an exemplary embodiment, the output component 206 can receive at least a portion of a source table from the search system 120, wherein the values of the attribute are included in the portion of the source table. The output component 206 includes a display component 208 that is configured to display at least the portion of the source table on a display of the client computing device 102. Display of the portion of the source table allows the user of the client computing device 102 to review contents of the source table prior to, for example, loading the attribute values of the source table into a spreadsheet.

[0043] As will be described in greater detail herein, the display component 208 can display a multitude of different parameters of the source table. In an exemplary embodiment, the display component 208 can receive a ranked list of search results generated by the search system 120, wherein the ranked list of search results includes a result that is representative of the above-mentioned source table. Pursuant to an example, the search results can include titles of respective source tables, locations of respective source tables (e.g., URL), identities of attributes in the source table, identities of entities in the source table, and the like. The display component 208, for instance, can receive an indication that the user has selected a search result that represents the source table from the ranked list of search results. Responsive to receiving such indication, the display component 208 can display more of the source table, can cause a web page that includes the source page to be loaded by a browser, etc.

[0044] The output component 206 can also include a preview generator component 210 that can generate a preview of a table identified by the search system 120 based upon the selected entities. As indicated above, the display component 208 can display a ranked list of search results that are representative of respective source tables. The preview generator component 210 can receive an indication that the user has performed a predefined action relative to a search result (e.g., such as hovering with a mouse pointer, pressing and holding with a finger, outputting a voice command). Detection of such action can cause the preview generator component 210 to display a preview of the source table represented by the search result to be presented on the display of the client computing device 102. The

user of the client computing device 102 can thus quickly obtain previews of source tables represented in the ranked list of search results.

[0045] In another example pertaining to the preview generator component 210, as indicated above, the preview generator component 210 can cause a preview of a source table identified by the search system 120 to be displayed on a display of the client computing device 102. Furthermore, the preview generator component 210 can generate the preview of the source table to include controls for manipulating data shown in the preview – for instance, these controls can facilitate sorting of values in the preview of the source table by column, by row, etc. Furthermore, filtering capabilities of conventional spreadsheet applications can be enabled by the preview generator component 210.

[0046] In still yet another example, the output component 206 can receive a plurality of table snippets from the server 104, wherein the snippets summarize respective source tables identified by the search system 120 as being relevant to the entities transmitted to the server 104 by the transmitter component 204. The snippets can include, for example, data that identifies a title of a table summarized by the snippet, a location of the source table summarized by the snippet, attributes included in the source table summarized by the snippet, and a portion of the source table that is representative of the contents of such source table. It can, therefore, be ascertained that, prior to causing the application 112 on the client computing device 102 to load attribute values for presentment to the user of the client computing device 102, the user can be provided with information that assists the user in ascertaining whether the contents of the source table are trustworthy, assists the user with selecting an attribute, etc.

[0047] In yet another example, the user can select entities without identifying an attribute, and the resultant query (constructed by the entity augmentor system 114 or the search system 120) can be constructed accordingly. The search system 120 of the server 104 can identify source tables that are relevant to the entities, wherein the identified source tables include values for several different attributes (e.g., attributes occurring most often with the entities as indicated in the index 124). Identities of the several attributes can be output by the search system 120 and received by the display component 208, which surfaces the several attributes to the user (e.g., displays such attributes on a display of the client computing device 102). For instance, the display component 208 can display the attributes to indicate that the attributes are selectable (e.g., hyperlink the attributes, cause buttons to be presented in graphical relation to the attributes, etc.). The user can select at least one attribute in the displayed attributes, wherein selection of the at least one attribute

is received by the input receiver component 202. The input receiver component 202 can then update the query to reflect selection of the attribute, or transmit the attribute to the search system 120, which then updates the query. The search system 120 may then search the index 124 based upon the selected entities and the selected attribute.

5 [0048] The output component 206 can further include an augmentor component 212 that augments the entities loaded in the application with values of the attribute selected by the user. In an example, the augmentor component 212 can cause a new column to be added to a table, and can populate the column with the values of the attribute. The augmentor component 212 aligns the attribute values with their respective entities. In another
10 example, the augmentor component 212 can generate a page (e.g., a web page, a word processing page, or the like) that includes the entities and the values of the attribute in tabular form, such that the user of the client computing device 102 is able to identify, for an entity in the entities, the value of the attribute for the entity.

[0049] With reference now to Fig. 3, an exemplary functional block diagram of the
15 search system 120 is illustrated. The search system 120 includes an entity receiver component 302 that receives entities (e.g. from the transmitter component 204 of the entity augmentor system 114). Additionally, the entity receiver component 302 can receive an attribute specified by a user of the client computing device 102. Pursuant to an example, the entity receiver component 302 can construct a query that is to be executed over the
20 index 124 responsive to receipt of the entities and (optionally) the attribute.

[0050] The search system 120 further comprises a search component 304 that is configured to search the index 124 based upon the query and return at least a portion of a source table indexed in the index 124 to the entity augmentor system 114. The search component 304 can utilize any suitable search algorithm to identify source tables related to
25 the query.

[0051] The search component 304 includes a table ranker component 306 that ranks tables identified by the search component 304 as being relevant to the query. In an exemplary embodiment, the table ranker component 306 can rank source tables based upon coverage of source tables with respect to the selected entities (represented in the
30 query), trustworthiness of a source where the source table is located, etc.

[0052] The search component 304 can further include a returner component 308 that returns at least a portion of a source table identified by the search component 304 and ranked by the table ranker component 306 to the entity augmentor system 114 of the client computing device 102. In an exemplary embodiment, the returner component 308 can

output a ranked list of search results, wherein the search results are representative of respective source tables are ranked in accordance with a ranking output by the table ranker component 306. The ranked list of search results can include hyperlinks, table snippets, etc. Furthermore, the returner component 308 can be configured to identify most relevant portions of source tables identified by the search component 304. For example, the returner component 308 can return a portion of a source table to the entity augmentor system 114, wherein the portion of the source table includes most popular attributes, attribute values believed to be most relevant to the selected entities, (e.g., based upon data previously accessed or manipulated by a user of the client computing device 102, attribute values already existent in a table loaded into the application 112, etc.).

[0053] The search component 304 may optionally include an attribute ranker component 310 that ranks attributes in source tables identified by the search component 304 based upon the query. The attribute ranker component 310 can select and rank attributes based upon one or more of the following: 1) attributes already represented in data loaded into the application 112 on the client computing device 108; 2) trustworthiness of the source table in which values of the attribute are included; 3) popularity of the attribute (e.g., based upon a number of tables indexed in the index 124 that include the attribute), or the like. The returner component 308 can return the ranked list of attributes to the entity augmentor system 114. The ranked list of attributes can be exposed or surfaced to a user of the client computing device 102, who may then select an attribute from the ranked list of attributes. The selected attribute may be transmitted to the server computing device 104 and received by the entity receiver component 302, which can update the query based upon the selected attribute. The search component 304 can then update the search using the updated query, such that source tables that include entities in the selected entities and values of the selected attribute are identified. The table ranker component 306 ranks the source tables, and the returner component 308 may then return a ranked list of search results (which may include respective portions of the source tables represented by the search results) to the entity augmentor system 114.

[0054] With reference collectively to Figs. 4-12, several exemplary graphical user interfaces that facilitate performance of a DFD search are illustrated. With reference solely to Fig. 4, an exemplary graphical user interface 400 of a spreadsheet application is illustrated. Thus, in the example shown in Fig. 4, the application 112 loaded into the client memory 110 can be a spreadsheet application. The graphical user interface 400 includes a spreadsheet 402 that comprises a plurality of cells arranged in rows and columns. In the

exemplary graphical user interface 400, columns A-F and rows 1-13 are depicted. It is to be understood that a graphical user interface of a spreadsheet application may depict more or fewer columns and/or rows than what is shown in the graphical user interface 400. The graphical user interface 400 also includes a toolbar 404 can include various selectable
5 buttons, pull-down menus, data entry fields, and the like, wherein such features facilitate creating, arranging, manipulating, filtering, and displaying data included in the spreadsheet 402.

[0055] The graphical user interface 400 further includes a query field 406 that is configured to receive a query set forth by a user of the client computing device 102. A
10 button 408, in graphical relation to the query field 406, when selected by a user of the client computing device 102, can cause a search to be performed over data based upon the query set forth in the query field 406. Pursuant to an example, the search can be performed locally (e.g., over data loaded into the spreadsheet application). In another example, a query entered into the query field 406 can identify several entities (and optionally an
15 attribute), and can be transmitted to the search system 120 responsive to the button 408 being selected. The graphical user interface 400 further includes a button 410 that, when selected, activates a microphone of the client computing device 102 to facilitate receipt of voice input. The voice input may be a natural language query, identification of a row or column in the spreadsheet 402, identification of entities represented in the spreadsheet, etc.

[0056] The graphical user interface 400 depicts that column A comprises several
20 entities. A plurality of entities 411 (entities in rows 1-8 of column A) are shown as being selected in the spreadsheet 402. For instance, a cursor 412 can be employed to select the plurality of entities 411. In another example, a gesture over a touch sensitive display can be employed to select the plurality of entities 411. In still yet another example, the
25 plurality of entities 411 can be selected via voice input (submitted responsive to selection of the button 410). In another example, the plurality of entities 411 can be selected by way of entry of a query into the query field 406.

[0057] Referring now to Fig. 5, an exemplary graphical user interface 500 that facilitates performance of a DFD search based upon the selected entities 411 is illustrated. The user
30 of the client computing device can set forth a command (e.g., a right-click, a gesture, a spoken command, etc.) that indicates that a DFD search is to be performed based upon the selected entities 411. The input receiver component 202 can receive such input and cause, for example, a pop-up window 502 to be displayed on a display of the client computing device 102 responsive to receipt of the command. The pop-up window 502 can include

several selectable options, wherein one of the options can initiate a DFD search. The option for initiating the DFD search can be selected by way of the cursor 412, by way of a touch on a touch-sensitive display, a voice command, etc. Responsive to the input receiver component 202 receiving the selection of the DFD option shown in the pop-up window 502, the transmitter component 204 can transmit the selected entities 411 to the search system 120 (or construct a query based upon the selected entities 411 and transmit the query to the search system 120). In this example, the user has not specified an attribute. It is to be understood, however, that mechanisms that facilitate receipt of a user-specification of an attribute are contemplated. For example, user can specify the attribute in the query field 406, the popup window 502 may include a field for specifying the attribute, etc.

[0058] Referring now to Fig. 6, a graphical user interface 600 is depicted that illustrates augmentation of the entities 411 with attribute values. As described above, the search system 120 searches the index 124 based upon the query (which is constructed based upon the entities 411 and a specified attribute), and can return values 602 of the attribute to the entity augmentor system 114. The augmentor component 212 can augment the spreadsheet 402 with the values 602 of the attribute by populating cells in column B of the spreadsheet 402 with the values 602 of the spreadsheet, appropriately aligned with the entities. Thus, “VALUE 1” is the value of the attribute for “ENTITY 1”, “VALUE 2” is the value of the attribute for “ENTITY 2”, and so on.

[0059] In the above-described example, the augmentor component 212 automatically populates the cells in column B with the values 602 of the attribute (e.g., the user has not reviewed and approved the values 602 prior to the values 602 being imported into the spreadsheet 402). If the user disapproves of the values 602 of the attribute, the user can set forth an “undo” command and cause a more refined search to be performed by the search system 120. Furthermore, in an example, the user need not specify the attribute. Rather, the search system 120 can automatically identify the attribute. For instance, the search system 120 can determine that “ATTRIBUTE 1” is the most popular attribute related to the selected entities 411 (e.g., based upon a number of tables indexed in the index 124 that include both entities in the selected entities 411 and “ATTRIBUTE 1”). The search component 304 can select such attribute without user feedback and transmit values for such attribute to the entity augmentor system 114, and the augmentor component 212 can augment the spreadsheet 402 with the attribute values responsive to receipt of such values.

[0060] To facilitate augmenting the spreadsheet with the values 602, the search component 304 and/or the augmentor component 212 can perform an integrated fuzzy-

joined column import. Specifically, the search component 304 and/or the augmentor component 212 can perform a fuzzy join between 1) the selected entities 411 in the spreadsheet; and 2) a column of the source table identified by the search component 304 that includes at least a subset of the selected entities 411. In an exemplary embodiment, such fuzzy join can be robust to spelling mistakes in different syntactic and semantic representations of the entities (either in the selected entities 411 or in the source table). For instance, the semantic synonyms may be retrieved from a separate device providing such synonymous entities. The search component 304 and/or the augmentor component 212 may also include syntactic transformation rules that are available when performing the fuzzy join. This fuzzy join allows for entities in source tables to be appropriately mapped to the selected entities 411, and thus appropriate attribute values to be mapped to the selected entities 411.

[0061] Now referring to Fig. 7, another exemplary graphical user interface 700 of the spreadsheet application is depicted. As shown in the graphical user interface 700, a user selects an attribute value from column B that was retrieved during the DFD search. For instance, the cursor 412 can be hovered over cell B5, which includes "VALUE 4" for "ENTITY 4". Responsive to the user hovering over the cell, information pertaining to the attribute value in the cell can be presented. For example, identity of the source table from which the attribute value was obtained can be presented in a pop-up window 702 responsive to the cursor 412 being hovered over cell B5. Further, the popup window 702 can display an identity of the entity in the source table that corresponds to "VALUE 4. This can indicate that the entity in the selected entities 411 is spelled differently in the source table. In an example, the entity in cell A5 may be "Company", while the corresponding entity in the source table may be spelled "CMPNY." Based upon the information in the pop-up window 702, the user of the client computing device 102 can verify correctness of the fuzzy join, correctness of individual attribute values, etc.

[0062] With reference now to Fig. 8, an exemplary graphical user interface 800 of the spreadsheet application is displayed. The graphical user interface 800 sets forth a depiction of data that can be presented on a display of the client computing device 102 responsive to a DFD search being conducted. The search system 120 returns search results 802 for presentment in the graphical user interface 800, and the display component 208 displays the search results 802 on the display of the client computing device 102. As indicated above, the search results 802 may include a ranked list of search results, wherein the search results are representative of source tables that have been identified by the search

system 120 as being relevant to a query (where the query is based upon the selected entities 411 and optionally at least one attribute specified by the user).

[0063] For instance, the search results 802 can be displayed in a manner similar to conventional web search results, where the search results 802 include hyperlinks to source tables respectively represented by the search results. Selection of a hyperlink can, for example, cause a web browser to load the web page that includes the source table. In another example, selection of the hyperlink can cause the preview generator component 210 to display a preview of the source table represented by the corresponding search result in the graphical user interface 800. While not shown, the search results 802 can have a scrollbar associated therewith that facilitates “infinite” scrolling. Thus, the user scrolls to be provided with additional search results, rather than requesting new pages of search results. In the exemplary graphical user interface 800, the search results 802 can include a plurality of ranked table snippets 804-810. The table snippets 804-810 respectively summarize source tables identified by the search system 120.

[0064] Now referring to Fig. 9, an exemplary table snippet 900 that can be included in the search results 802 is illustrated. The table snippet 900 can include, but is not limited to including: 1) a source table title; 2) a location of the source table summarized by the table snippet 900 (e.g., a URL where the table can be acquired); 3) attributes having attribute values in the source table summarized by the table snippet 900 (e.g., a threshold number of most popular attributes, a threshold number of attributes believed to be most relevant to the selected entities 411, and/or attributes submitted by the user of the client computing device 102); 4) entity coverage data, wherein the entity coverage data is indicative of a number of entities in the selected entities 411 that are also included in the source table summarized by the table snippet 900; and 5) a sample portion 902 of the source table summarized by the table snippet 900, wherein the sample portion 902 of the source table includes a subset of rows and/or columns of the source table that can provide the user of the client computing device 102 with information about why the search system 120 identified the source table based upon the selected entities 411. For instance, rows and/or columns can be included in the portion 902 that comprise entities in the selected entities 411 and/or attribute values for such entities.

[0065] It is to be understood that the rows and/or columns in the portion 902 of the source table may be arranged in a differently when compared to the source table itself. That is, the search system 120 can generate the table snippet 900 to appear to correspond to the arrangement of entities in the spreadsheet loaded in the client computing device 102.

Accordingly, the user of the client computing device 102 can relatively quickly visually ascertain why the search system 120 returned the source table summarized by the snippet 900, and can further visually ascertain if it is desirable to augment the spreadsheet 402 with attribute values in the source table summarized by the snippet.

5 **[0066]** In an exemplary embodiment, the attributes listed in the table snippet 900 can be selectable. Upon receipt of a selection of an attribute, the table snippet 900 can be updated such that the portion 902 highlights values of the attribute relative to entities (e.g., entities included in the source table that are also included in the selected entities 41). In another example, selection of an attribute can cause the query to be refined, and the search system
10 120 to perform an updated search over the index 124.

[0067] Furthermore for each row and/or column of the portion 902 of the source table summarized by the table snippet 900, a respective graphical object can be presented, wherein selection of such graphical object can cause the augmentor component 212 to augment the spreadsheet 402 with values of the attribute in the respective row and/or
15 column. For example, the portion 902 of the source table illustrated in the table snippet 900 includes five columns: a first column that is representative of entities in the source table summarized by the table snippet 900; and four columns that are representative of respective attributes of the entities. Each of the four attribute columns can include a respective selectable graphical object. Thus, the table snippet 900 includes selectable
20 graphical objects 904-910 corresponding to the four attribute columns. In an example, responsive to receipt of a selection of the graphical object 904, attribute values, of the attribute “ATTRIBUTE 1”, of entities in the selected entities 411 can be loaded into the spreadsheet 402 and graphically depicted in relation to the selected entities 411 (for example, as shown in the graphical user interface 600 of Fig. 6). That is, a column of the
25 spreadsheet 402 can be newly populate an attribute column for “ATTRIBUTE 1”, where the column is populated by attribute values.

[0068] The table snippet 900 may also include an expand button 912 that, when selected, may cause an entirety of the source table summarized by the snippet 900 to be presented on the display of the client computing device 102 (e.g., in the spreadsheet
30 application, in a web browser, in a word processing document, etc.) into For example, the source table can be loaded into a new spreadsheet of the spreadsheet application. In another example, the search results 802 can be replaced with the table summarized by the table snippet 900. Further, while not shown, the table snippet 900 can include a search button that is configured to cause the search system 120 to identify source tables similar to

the source table summarized by the table snippet 900. The search system 120 may then provide search results that are based upon the source table summarized by the table snippet 900 (and optionally, further based upon the selected entities 411 and an attribute set forth by the user of the client computing device 102).

5 [0069] Moreover, while not shown, the table snippet 900 can support hit highlighting. With more particularity, a table represented by the table snippet 900 is retrieved based upon a query, wherein the query includes keyword(s). The keywords, where included in the table snippet 900, can be highlighted to indicate where in the table represented by the table snippet 900 the keywords occur. The locations can be in the title, the URL, the subset
10 of the table that is included in the snippet 900, etc.

[0070] With reference now to Fig. 10, another exemplary graphical user interface 1000 that facilitates performance of a data finding data search. The exemplary graphical user interface 1000 can be presented by an application executed by the client computing device 102 other than a spreadsheet application. For example, the graphical user interface 1000
15 may be for a word processing application, a text editing application, an e-mail application, a social networking application, an instant messaging application, a news application, a slide presentation application, or other suitable application that can present content on a page. As indicated, the content on the page includes several entities 1001 (e.g., separated by delimiters). Any suitable selection mechanisms can be employed to select the entities
20 1001 (e.g., entities 1-6) of the content of the content page. For example, the user of the client computing device 102 can highlight the entities 1001 through utilization of the cursor 412. In another example, the user can select the entities 1001 by way of a voice command, a gesture, or the like. The user may then set forth an indication that a data finding data search is to be employed based upon the selected entities 1001 (and
25 optionally, a provided attribute). Again, this indication can be received by, for instance, right-clicking on the selected entities 1001, which causes a pop-up window 1002 to be presented on a display of the client computing device 102.

[0071] One of the options in the pop-up window 1002 can be to perform a DFD search based upon the selected entities 1001, and such option can be selected by way of any
30 suitable selection mechanism. The selected entities 1001 (or a query constructed based upon the selected entities 1001) can then be transmitted to the search system 120, which can return search results, attribute values, etc., in a manner similar to what has been described above. In an exemplary embodiment, the search system 120 can cause a new page to be constructed (e.g., a web page, a word document, etc.) and can cause the new

page to be transmitted to the client computing device 102, wherein the resultant page includes a table that comprises the selected entities 1001 and corresponding attribute values identified by the user and/or the search system 120. In another example, responsive to receipt of attribute values for the selected entities 1001 from the search system 120, a spreadsheet application can be initiated at the client computing device 102 and populated with the selected entities 1001 and attribute values corresponding thereto. In such an embodiment, the user can identify a table of interest on a web page, can select entities in the table of interest, initiate performance of a DFD search using the selected entities (and optionally an attribute), and cause a table to be automatically generated in a spreadsheet application based upon the DFD search (where the table includes the entities selected by the user and attribute values from another table identified by the search system 120 and approved by the user).

[0072] With reference now to Fig. 11, another exemplary graphical user interface 1100 is depicted. The graphical user interface 1100 can be a graphical user interface for a search engine (e.g., a particular vertical of a search engine). For example, the search engine may include a plurality of verticals, which can be represented by selectable graphical objects in the graphical user interface 1100. The exemplary verticals can be “WEB”, “IMAGES”, “VIDEOS”, “NEWS”, and “TABLES” represented by selectable objects 1102-1110. The “WEB” vertical facilitates performance of conventional web searches, the “IMAGES” vertical facilitates performance of searches for images, the “VIDEOS” vertical facilitates performance of searches for videos, and so forth.

[0073] In the exemplary graphical user interface 1100, the user has selected the graphical object 1110 with the cursor 412, such that the “TABLES” vertical has been selected. The “TABLES” vertical can include at least two fields that can be populated: 1) a first field 1112 that is configured to receive an entity or entities specified by the user; and 2) a second field 1114 that is configured to receive an attribute or attributes of the entity or entities entered into the first field 1112. As indicated, the user can enter multiple entities into the field 1112, separated by delimiters (e.g., semicolons or other suitable delimiters). Likewise, the user can set forth multiple attributes in the field 1114, separated by delimiters. A search button 1116 in the graphical user interface 1100 may then be selected, which causes the transmitter component 204 to transmit the entities and attributes in the fields 1112 and 1114, respectively, to the search system 120 (or which causes the entity augmentor system 114 to construct a query based upon the entities and attributes and

transmit the query to the search system 120). The search system 120 can execute a DFD search in the manner described above.

[0074] The search system 120 may then return search results, which can be displayed in a search results field 1118. The search results shown in the field 1118 may be portions of source tables identified as including at least one entity set forth in the field 1112 and at least one value of the attribute set forth in the field 1114. In another example, the search results in the search results field 1118 can include data extracted from multiple tables that are joined together to form a final result that can be relatively quickly consumed by the user. For example, a first source table may include a first entity specified in the field 1112 and have an attribute value for the attribute specified in the field 1114, while a second source table may include a second entity specified in the field 1112 and a value for the attribute specified in the field 1114. The search system 120 can identify the first source table and the second source table, select appropriate entries from each of the two source tables, and create a search result that is based upon the combination of the first source table and the second source table. The information need of the user is met by the combination included in the search result.

[0075] Now referring to Fig. 12, another exemplary graphical user interface 1200 is illustrated that facilitates performance of a DFD search. The graphical user interface 1200 includes a first portion 1202 that can include a plurality of selectable objects 1204-1206.

For example, such objects 1204-1206 may be included on a web page, wherein the objects 1204-1206 can represent purchasable items. The user may scroll through the objects 1204-1206, select an object or objects by way of a suitable selection mechanism, and drag the selected object(s) to a workspace field 1208. In the exemplary graphical user interface 1200, the user has selected the object 1204, as well as three additional objects 1210-1214. The workspace field 1208 may include a search button 1216 that, when selected by the user, causes the transmitter component 204 to transmit the entities in the workspace field 1208 to the search system 120, which executes a search over the index 124 based upon the entities, as described above. Search results may be presented in the workspace field 1208 or on another content page.

[0076] While the graphical user interfaces depicted in Figs. 4-12 provide various examples, it is to be understood that numerous variants are contemplated. For example, while the graphical user interface 1100 depicts two fields for initiating performance of a DFD, it is to be understood that a DFD search may be initiated through utilization of a single field (e.g., the query field 406, the field 1112, etc.). In such an example, a keyword query may

be entered into the single field, and the search system 120 can return a ranked list of tables based upon the keyword query. The query set forth in the field may include implicit structure, and there may be certain classes of queries for which the search system 120 is optimized. For example, structures that are exemplary include a pure keyword query, an entity only query, a multiple entity query, a concept query, an entity-attribute query, and the like.

[0077] Further, while the graphical user interface 700 depicted in Fig. 7 illustrates data that can be presented upon selecting a cell, it is be understood that user may wish to obtain information about a particular column. For instance, the cursor 412 can be employed to hover over the “B” column in the graphical user interface 700, which may cause the pop-up window 702 to present metadata about the column, such as column synonyms. Such synonyms may be existent in source tables indexed in the index 124.

[0078] In yet another example, the exemplary table snippet 900 has been described as including the graphical objects 904-910, wherein when one of the graphical objects 904-910 is selected, attribute values in the column of the selected object are appended to a table in a loaded spreadsheet. In another example, however, a drag-and-drop approach to append data is contemplated, wherein the user selects a particular column or cell values in the table snippet 900 (or expanded table), drags the column or cell values into the spreadsheet 402 loaded by the spreadsheet application at a desired location, and drops the column or cell values in at a desired position(s). When the drag and drop of attribute values is performed, the attribute values can be automatically arranged to correspond to the appropriate entities.

[0079] Further, an embodiment has been described herein where the search system 120 returns a ranked list of attributes. In another example, rather than being provided with attributes in list form, the display component 208 can display a tag cloud of attributes for the selected entities 411, where the attributes (or key words from attributes) for the entities 411 are included in the tag cloud. The attributes in the tag cloud can be sized in proportion to the number of identified source tables that comprise the attribute name and/or keywords from the attribute names (in the case of an entity-attribute query). Other examples will be readily recognized.

[0080] Figs. 13-17 illustrate exemplary methodologies relating to DFD searches. While the methodologies are shown and described as being a series of acts that are performed in a sequence, it is to be understood and appreciated that the methodologies are not limited by the order of the sequence. For example, some acts can occur in a different order than

what is described herein. In addition, an act can occur concurrently with another act. Further, in some instances, not all acts may be required to implement a methodology described herein.

[0081] Moreover, the acts described herein may be computer-executable instructions that can be implemented by one or more processors and/or stored on a computer-readable medium or media. The computer-executable instructions can include a routine, a sub-routine, programs, a thread of execution, and/or the like. Still further, results of acts of the methodologies can be stored in a computer-readable medium, displayed on a display device, and/or the like.

[0082] Now referring to Fig. 13, an exemplary methodology 1300 that facilitates performance of a DFD search is illustrated. The methodology 1300 starts at 1302, and at 1304 computer readable entities are received, wherein the computer readable entities have a common attribute. In an example, the computer-readable entities can be received responsive to a user selecting the entities by way of a suitable selection mechanism. At 1306, responsive to receiving the entities, the entities are transmitted to a search system. As indicated above, the search system is configured to search over an index that indexes source tables, wherein the search is based upon the entities.

[0083] At 1308, values of the common attribute are received from the search system, wherein the values respectively correspond to the received entities. At 1310, responsive to receiving the values of the common attribute, the values of the common attribute are output. For example, the values of the common attribute can be presented as a portion of a search result. In another example, the values of the common attribute can be appended to a table that includes the entities, wherein the values of the attribute are arranged in the table to respectively appropriately correspond to the entities. The methodology 1300 completes at 1312.

[0084] Now referring to Fig. 14, an exemplary methodology 1400 that facilitates augmenting tabular data with attribute values retrieved based upon a DFD search is illustrated. The methodology 1400 starts at 1402, and at 1404 a plurality of computer readable entities are received, wherein the entities share an attribute. At 1406, responsive to receiving the entities, the entities are transmitted to a search system, wherein the search system is configured to search over an index of computer-readable source tables. At 1408, a plurality of table snippets are received from the search system, wherein the table snippets are respective summaries of source tables identified by the search system. At 1410, the plurality of table snippets are displayed on the display. At 1412, a selection of a table

snippet is received from the plurality of table snippets, and at 1414 the entities are augmented (enriched) with values of the attribute, wherein the values are extracted from a source table summarized by the table snippet and are retrieved based upon the selection of the table snippet at 1412. The methodology 1400 completes at 1416.

5 [0085] Now referring to Fig. 15, an exemplary methodology 1500 that facilitates augmenting entities with attribute values based upon selection of an attribute in a ranked list of attributes (or a tag cloud of attributes) is illustrated. The methodology 1500 starts at 1502, and at 1504 entities are received, wherein the entities have a common attribute. At 1506, responsive to receiving the entities, the entities are transmitted to a search system
10 that is configured to search over an index of computer readable tables. At 1508, attributes are received from the search system, wherein the attributes are identified as being relevant to the entities received at 1504. In an example, the attributes can be received as a ranked list of attributes, as a tag cloud of attributes, etc. For example, the attributes can be attributes that most commonly occur in source tables (indexed by the index) that include
15 the entities received at 1504.

[0086] At 1510, responsive to receiving the attributes, the attributes are output. At 1512, a selection of an attribute is received (e.g., a user selects an attribute of interest), and at 1514, responsive to receiving the selection of the attribute, the entities are augmented with respective values of the attribute. With more particularity, selection of the attribute can
20 cause the attribute to be transmitted to the search system, which utilizes the attribute and the received entities to locate attribute values in source tables indexed by the index. The attribute values can then be returned and confirmed by the user, and tabular data can be augmented with the attribute values. The methodology 1500 completes at 1516.

[0087] Now referring to Fig. 16, an exemplary methodology 1600 that facilitates
25 dynamic generation of table snippets is illustrated. The methodology 1600 starts to 1602, and at 1604 computer-readable entities are received. At 1606, responsive to receiving the computer-readable entities, a search for computer-readable source tables is executed based upon the entities. That is, source tables are located that include the entities (or synonyms thereof). Furthermore, the search can be executed based upon an identified attribute. At
30 1608, a ranked list of tables is generated based upon the search, and at 1610 table snippets for respective tables in the ranked list of tables are generated. At 1612, the table snippets are output. For example, the table snippets can be displayed in a graphical user interface of a spreadsheet application. The methodology 1600 completes 1614.

[0088] Now referring to Fig. 17, an exemplary methodology 1700 that facilitates generation of a ranked list of attributes is illustrated. The methodology 1700 starts at 1702, and at 1704 computer readable entities are received. At 1706, responsive to receiving the entities, a search is executed for computer-readable tables that include the entities. At 1708, a ranked list of attributes related to the entities is generated based upon the search. At 1710, the ranked list of attributes is output (e.g., transmitted to a client computing device). At 1712, a selection of an attribute from the ranked list of attributes is received, and at 1714 values of the attribute for the respective entities are output responsive to receipt of the selection of the attribute. The methodology 1700 completes at 1716.

[0089] Referring now to Fig. 18, a high-level illustration of an exemplary computing device 1800 that can be used in accordance with the systems and methodologies disclosed herein is illustrated. For instance, the computing device 1800 may be used in a system that facilitates selection of computer-readable entities and/or attributes for use when performing a DFS search. By way of another example, the computing device 1800 can be used in a system that supports performing a search based upon identified entities and attributes. The computing device 1800 includes at least one processor 1802 that executes instructions that are stored in a memory 1804. The instructions may be, for instance, instructions for implementing functionality described as being carried out by one or more components discussed above or instructions for implementing one or more of the methods described above. The processor 1802 may access the memory 1804 by way of a system bus 1806. In addition to storing executable instructions, the memory 1804 may also store tabular data, computer-readable entities, attribute names, an index of source tables, etc.

[0090] The computing device 1800 additionally includes a data store 1808 that is accessible by the processor 1802 by way of the system bus 1806. The data store 1808 may include executable instructions, source tables, an index of source tables, computer-readable entities, etc. The computing device 1800 also includes an input interface 1810 that allows external devices to communicate with the computing device 1800. For instance, the input interface 1810 may be used to receive instructions from an external computer device, from a user, etc. The computing device 1800 also includes an output interface 1812 that interfaces the computing device 1800 with one or more external devices. For example, the computing device 1800 may display text, images, etc. by way of the output interface 1812.

[0091] It is contemplated that the external devices that communicate with the computing device 1800 via the input interface 1810 and the output interface 1812 can be included in

an environment that provides substantially any type of user interface with which a user can interact. Examples of user interface types include graphical user interfaces, natural user interfaces, and so forth. For instance, a graphical user interface may accept input from a user employing input device(s) such as a keyboard, mouse, remote control, or the like and provide output on an output device such as a display. Further, a natural user interface may enable a user to interact with the computing device 1800 in a manner free from constraints imposed by input device such as keyboards, mice, remote controls, and the like. Rather, a natural user interface can rely on speech recognition, touch and stylus recognition, gesture recognition both on screen and adjacent to the screen, air gestures, head and eye tracking, voice and speech, vision, touch, gestures, machine intelligence, and so forth.

[0092] Additionally, while illustrated as a single system, it is to be understood that the computing device 1800 may be a distributed system. Thus, for instance, several devices may be in communication by way of a network connection and may collectively perform tasks described as being performed by the computing device 1800.

[0093] Various functions described herein can be implemented in hardware, software, or any combination thereof. If implemented in software, the functions can be stored on or transmitted over as one or more instructions or code on a computer-readable medium. Computer-readable media includes computer-readable storage media. A computer-readable storage media can be any available storage media that can be accessed by a computer. By way of example, and not limitation, such computer-readable storage media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to carry or store desired program code in the form of instructions or data structures and that can be accessed by a computer. Disk and disc, as used herein, include compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk, and Blu-ray disc (BD), where disks usually reproduce data magnetically and discs usually reproduce data optically with lasers. Further, a propagated signal is not included within the scope of computer-readable storage media. Computer-readable media also includes communication media including any medium that facilitates transfer of a computer program from one place to another. A connection, for instance, can be a communication medium. For example, if the software is transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio and microwave

are included in the definition of communication medium. Combinations of the above should also be included within the scope of computer-readable media.

[0094] Alternatively, or in addition, the functionally described herein can be performed, at least in part, by one or more hardware logic components. For example, and without
5 limitation, illustrative types of hardware logic components that can be used include Field-programmable Gate Arrays (FPGAs), Program-specific Integrated Circuits (ASICs), Program-specific Standard Products (ASSPs), System-on-a-chip systems (SOCs), Complex Programmable Logic Devices (CPLDs), etc.

[0095] What has been described above includes examples of one or more embodiments.

10 It is, of course, not possible to describe every conceivable modification and alteration of the above devices or methodologies for purposes of describing the aforementioned aspects, but one of ordinary skill in the art can recognize that many further modifications and permutations of various aspects are possible. Accordingly, the described aspects are intended to embrace all such alterations, modifications, and variations that fall within the
15 spirit and scope of the appended claims. Furthermore, to the extent that the term “includes” is used in either the details description or the claims, such term is intended to be inclusive in a manner similar to the term “comprising” as “comprising” is interpreted when employed as a transitional word in a claim.

CLAIMS

1. A computing system comprising:
a processor; and
a memory that comprises an application that is executed by the processor, the application comprises an entity augmentor system, the entity augmentor system comprises:
an input receiver component that receives computer-readable entities and transmits the entities to a search system that is configured to search over an index of source tables based upon the entities, wherein an entity is a string that represents a person, place, or thing, and wherein people, places, or things respectively represented by the entities have an attribute; and
an output component that, responsive to the input receiver component receiving the entities, receives respective attribute values for the attribute from the search system, the attribute values respectively correspond to the entities, and the output component outputs the attribute values responsive to receiving the attribute values.
2. The computing system of claim 1 further comprising a display that is in communication with the processor, the application being a spreadsheet application, the spreadsheet application causes a spreadsheet to be presented on the display, the spreadsheet comprises a plurality of fields that are respectively populated by the entities, and wherein the output component comprises an augmentor component that augments the spreadsheet with the attribute values.
3. The computing system of claim 2, the augmentor component, when augmenting the spreadsheet, populates one of a row or a column of the spreadsheet with the attribute values.
4. The computing system of claim 1 further comprising a display that is in communication with the processor, the application being one of a web browser or an email application, the entities are included in a web page being displayed on the display or an email being displayed on the display, the input receiver component receives the entities responsive to receipt of a selection of the entities from the one of the web page or the email.

5. The computing system of claim 1 further comprising a display, the output component comprises a display component that receives a plurality of table snippets from the search system, the table snippets summarizing respective source tables identified by the search system as being relevant to the entities, the display component displays the table snippets on the display.

6. The computing system of claim 5, the output component further comprises an augmentor component, the input receiver component receives a selection of a table snippet in the table snippets, the attribute values comprised by a source table summarized by the table snippet, and wherein the augmentor component augments the entities with the respective attribute values responsive to receipt of the selection of the table snippet.

7. The computing system of claim 6, the augmentor component receives an indication that a column represented in the table snippet has been selected from the table snippet, the augmentor component adds contents of the column to a table presented by an application being executed by the processor.

8. The computing system of claim 1, the input receiver component further receives a selection of the attribute, the input component transmits the identity of the attribute to the search system with the entities.

9. A method executed at a client computing device, the method comprising:
receiving machine-readable entities, an entity being a string that is representative of a person, place, or thing, and wherein people, places, or things represented by the entities have an attribute;

responsive to receiving the entities, transmitting the entities to a search system, wherein the search system is configured to perform a search over an index of network-accessible tables based upon the entities;

receiving, from the search system, respective attribute values for the entities in the list of entities, the attribute values being values of the attribute; and

responsive to receiving the attribute values, displaying the attribute values on a display, wherein the attribute values are respectively displayed in graphical relation to the entities to visually depict that the people, places, or things represented by the entities have the respective attribute values.

10. The method of claim 9, wherein receiving the entities comprises:
displaying the entities in a graphical user interface of an application that is being executed by the client computing device; and
receiving a selection of the entities by way of an interaction with the graphical user interface.
11. The method of claim 9, the interaction being one of:
a selection the entities by way of a mouse pointer;
a selection of the entities by way of a gesture performed on a touch-sensitive display of the client computing device; or
a spoken command that identifies the entities.
12. The method of claim 9, wherein receiving the entities comprises:
displaying tabular data in a graphical user interface of a spreadsheet application that is being executed by the client computing device, the tabular data presented in cells of the graphical user interface, a subset of cells in the cells respectively comprise the entities;
receiving a selection of the subset of cells; and
receiving an indication that a search is to be performed based upon the subset of the cells; wherein displaying the attribute values on the display comprises augmenting the tabular data with a new row or column, the new row or column populated with the attribute values.
13. The method of claim 9, wherein responsive to transmitting the entities to the search system, attributes are received from the search system, the attributes comprise the attribute, the method further comprises:
receiving a selection of the attribute from the attributes;
responsive to receiving the selection of the attribute from the attributes,
transmitting the attribute to the search system; and
receiving the attribute values from the search system responsive to transmitting the attribute to the search system.

14. The method of claim 9, further comprising:

responsive to transmitting the entities to the search system, receiving a ranked list of table snippets from the search system, each table snippet in the ranked list of table snippets summarizes a respective source table accessible by way of a network connection, each table snippet in the ranked list of table snippets comprises:

a title for the respective source table;

a network location of the respective source table; and

a representative portion of the respective source table;

displaying the ranked list of table snippets on the display;

receiving a selection of a table snippet in the ranked list of table snippets, the table snippet summarizes a source table that comprises the attribute values;

responsive to receiving the selection of the table snippet, transmitting the selection of the table snippet to the search system; and

receiving the attribute values responsive to transmitting the selection of the table snippet to the search system.

15. A computer-readable storage medium comprising instructions that, when executed by a processor, cause the processor to perform acts comprising:

displaying a spreadsheet in a graphical user interface of a spreadsheet application, the spreadsheet comprises a table, the table comprises a column that includes entities;

receiving a selection of the entities in the table;

receiving an indication that a search is to be conducted based upon the entities selected in the table;

receiving an identity of an attribute;

transmitting the entities and the attribute to a search system, the search system configured to execute a search for computer-readable tables based upon the entities and the attribute;

responsive to transmitting the entities and the attribute to the search system, receiving values of the attribute that respectively correspond to the entities; and

responsive to receiving the values of the attribute, adding another column to the table in the spreadsheet, the another column comprises the values of the attribute, the values of the attribute arranged in the another column to respectively map to the entities in the column.