SYSTEM AND METHOD FOR EFFICIENT REPRESENTATIONS OF EMAIL CONVERSATIONS AND USE OF SUCH REPRESENTATIONS IN SCOPING EMAIL CONVERSATIONS

Abstract

Embodiments as disclosed include document analysis systems that may obtain email data collected or obtained from email servers from one or more source systems and build a graph of the emails, where the nodes of the email graph represent data about an email and the edges in the graph between the nodes of the graph are determined based on metadata associated with the emails or the text content of the emails. These email graphs may be utilized to quickly and efficiently scope email conversations, including determining emails for a conversation that are subsumed by, or inclusive of, other emails in a conversation.

Inventors: **Bucao; Levi Jonathan (Austin, TX), Lee; Peter Anthony (Pflugerville, TX), Caramanica; Jamie Daniel (Shrewsbury, MA), Levinson; Liad (Austin, TX)**

Applicant: **CS Disco, Inc.** (Austin, TX)

Family ID: **1000008586723**

Appl. No.: **19/186248**

Filed: **April 22, 2025**

Related U.S. Application Data

parent US continuation 18419826 20240123 parent-grant-document US 12265575 child US 19059902
parent US continuation 17187116 20210226 parent-grant-document US 11928154 child US 18419826
parent US continuation-in-part 19059902 20250221 PENDING child US 19186248
us-provisional-application US 63640427 20240430

Publication Classification

Background/Summary

RELATED APPLICATIONS [0001] This application claims a benefit of priority under 35 U.S.C. § 119 of the filing date of U.S. Provisional Patent Application No. 63/640,427 filed Apr. 30, 2024, entitled, "SYSTEM AND METHOD FOR EFFICIENT REPRESENTATIONS OF EMAIL CONVERSATIONS" which is fully incorporated by reference herein for all purposes. [0002] This application is a continuation-in-part of, and claims a benefit of priority under, 35 U.S.C. § 120 of the filing date of U.S. patent application Ser. No. 19/059,902 filed Feb. 21, 2025, entitled, "SYSTEM AND METHOD FOR EFFICIENT CREATION AND INCREMENTAL UPDATING OF REPRESENTATIONS OF EMAIL CONVERSATIONS" which is a continuation of U.S. patent application Ser. No. 18/419,826 filed Jan. 23, 2024, issued as U.S. Pat. No. 12,265,575, entitled, "SYSTEM AND METHOD FOR EFFICIENT CREATION AND INCREMENTAL UPDATING OF REPRESENTATIONS OF EMAIL CONVERSATIONS," is a continuation of, and claims a benefit of priority under 35 U.S.C. § 120 of the filing date of U.S. patent application Ser. No. 17/187,116 filed Feb. 26, 2021, issued as U.S. Pat. No. 11,928,154, entitled, "SYSTEM AND METHOD FOR EFFICIENT CREATION AND INCREMENTAL UPDATING OF REPRESENTATIONS OF EMAIL CONVERSATIONS," which are fully incorporated by reference herein for all purposes.

COPYRIGHT NOTICE

TECHNICAL FIELD

[0004] This disclosure relates generally to analysis and understanding of electronic documents. In particular, this disclosure relates to the analysis and understanding of emails. More specifically, this disclosure relates to the efficient creation and update of representation of email conversations in document analysis systems. Even more specifically, this disclosure relates to the dynamic threading of email conversation to scope these email conversations, and representing email conversations in document analysis systems to support or reflect such dynamic threading.

BACKGROUND

[0005] In the modern world, the vast majority of documents that are being created, utilized, and maintained, are in electronic format. A number of different situations commonly arise that require an analysis or identification of certain relevant electronic documents from a relatively large pool of available electronic documents. These situations are generally referred to as information retrieval or search problems. These types of search problems crop up in a wide variety of contexts. For example, in litigation, an entity's documents may need to be reviewed in order to identify documents that may be relevant to one or more issues in the litigation. In other examples, certain regulatory filings may require review of a number of documents to identify documents that may be relevant to one or more issues in the regulatory filing.

[0006] To illustrate in more detail, parties to litigation typically have to share relevant evidence with opposing counsel through the discovery process. In many cases, each party makes a reasonable search of their records based on some set of terms or keywords and produces the results of the search to the other party. Discovery thus typically involves the gathering of potentially relevant materials, much of it digital, and then reviewing such materials to determine what to be shared with opposite parties. Additionally, during the course of the litigation each party may continually review those documents produced by the opposing party to locate documents relevant to the case at hand. Litigation thus represents a microcosm of a more general problem raised by the high volume of electronic documents present in a variety of contexts. Namely, how can a large volume of electronic documents be understood, reviewed, or searched in order that documents relevant to a particular topic or user's interest may be located.

[0007] To aid users in resolving these problems, a document analysis system may be provided in a given electronic context. A document analysis system is a computer system used to process a corpus of electronically stored information and allow users to analyze, review or navigate the information, or search the electronic information to return electronically stored information responsive to a search.

[0008] Emails are a problematic subset of the documents analyzed by such a document analysis system as the nature of emails usually results in computationally intensive and slow analysis of such emails. In particular, in many cases emails may be substantially cumulative of one another, such as in the case where emails of an email conversation may quote or otherwise encompass previous emails in the conversation.

[0009] It would thus be desirable to have improved systems and methods for document analysis of email data that can, for example, efficiently determine emails that are subsumed by other emails and represent such email conversations accordingly such that these subsumed emails can be readily identified and emails of a group culled or presented accordingly.

SUMMARY

[0010] As discussed, in most contexts documents are now almost universally in electronic format. It is often the case that identification or other analysis of relevant electronic documents is required. For instance, regulatory filings may require review and identification of certain documents. Another primary example is litigation, where parties to litigation have to share relevant evidence through the discovery process. Discovery typically involves the gathering and sharing of potentially relevant materials, much of it digital documents, and then reviewing such materials to determine which documents are to be shared. Additionally, during the course of the litigation each party may continually review those documents produced by the opposing party to locate relevant documents. Litigation thus represents a specific instance of a more general problem raised by the high volume of electronic documents present in a variety of contexts. Specifically, how can a large volume of electronic documents be understood, reviewed, or searched in order that documents relevant to a particular topic or user's interest may be located.

[0011] To aid users in resolving these problems, a document analysis system may be provided in a given electronic context. A document analysis, document review, information retrieval, or search system (which all will be utilized here substantially interchangeably) is a computer system to process a corpus of electronically stored information (referred to as the corpus) and allow users to analyze, review or navigate the information, or search the electronic information to return electronically stored information responsive to a search (also referred to as a query). Items of electronic information that form a corpus may be referred to interchangeably as (electronic) documents, items, files, objects, items, content, etc. and may include objects such as files of almost any type including documents for various editing applications, emails, workflows, etc.

[0012] Emails are a problematic subset of the documents analyzed by such a document analysis system as the nature of emails usually results in computationally intensive and slow analysis of such emails. To continue with this discussion, a document analysis system processes a corpus of

electronically stored information including electronic documents and allows users to analyze, review, navigate or otherwise access or manipulate these documents or associated data. Emails are a problematic subset of the documents analyzed by such a document analysis system. Emails are ubiquitous within almost all modern enterprises, serving as the almost de-facto form of communication in such enterprises. It is thus often required to obtain and review such emails in the course of a litigation or other type of document review. As such, a document analysis system may provide a mechanism by which users can be presented with these emails and review such emails and associated data.

[0013] In many cases it is desirable for document analysis systems to identify and present such emails in the form of "conversation trees", where each conversation tree represents a conversation comprised of a set of associated emails that include an original or root email, (the beginning of the conversation) and all of the subsequent replies and forwards transitively associated with that original email.

[0014] The determination of such email conversations from email data in the document analysis environments is often times complex. Email data may comprise email metadata, or the actual emails document themselves, of a large volume of emails (e.g., on the order of hundreds of thousands or millions in some cases). Construction of the conversation trees requires the determination of associated emails and the original or root email of such a conversation tree from a large volume of email data. The processing of such email data may thus be computing time and resource intensive.

[0015] Complicating these determinations is the consideration that email data may be retrieved from different email servers (e.g., of different types) at different times.

[0016] Thus, the conversation trees determined at one point may be incomplete and new email data regarding such conversation trees may be obtained at (e.g., multiple) subsequent times. Accordingly, such document analysis systems may be required to frequently identify conversation trees pertaining to newly obtained email data and update the identified conversation trees. Because this newly obtained data may be less voluminous (e.g., than the number of emails already processed), users may expect that these incremental additions may be ingested quite quickly. However, typical methods for the incremental identification and update of such conversation trees may have less then desirable performance (e.g., from a computing time and resource standpoint).

[0017] Other factors also complicate the document analysis of emails in such document analysis systems. Namely, if such emails are obtained from multiple email servers for multiple user accounts of various users in an enterprise, there is likely to be a significant degree of duplicity of the emails between users. Duplicity may occur, for example, because the same email is sent to a number of recipients at once or because it is often the case that emails quote or include the text of previous emails (e.g., when responding or forwarding an email) and therefore may contain large amounts of duplicative content.

[0018] From a user's perspective it is desirable that such emails including duplicative content are identified by the document analysis system such that the user may review such duplicative content only once. In other words, if an inclusive email (e.g., and any included or subsumed emails) are identified to a user, the user may review all that content in the identified inclusive email and avoid having to review other emails identified as included in an inclusive email. The identification of such inclusive (or subsumed) emails is, however, not a trivial task, being quite computationally expensive, especially when such incremental updates to the email data as discussed above are taken into account.

[0019] It would thus be desirable to have improved systems and methods for document analysis of email data that can efficiently construct and incrementally update representations of email conversations. More particularly, it would be desirable to have improved systems and methods for document analysis of email data that can, for example, efficiently determine emails that are subsumed by (or inclusive of) other emails and represent such email conversations accordingly

such that these subsumed emails can be readily identified and emails of a group culled or presented accordingly.

[0020] To those ends, among others, attention is directed to embodiments of the document analysis systems disclosed herein. Embodiments of such document analysis systems may obtain email data on emails from source systems and organize and analyze such emails to allow users to review such email using the system. Such a document analysis system may be, for example, a multi-tenant platform that provides document analysis services to users from multiple (e.g., distinct) tenants, including allowing users of these tenants to access and review emails and email data associated with the tenant. For example, in a litigation context such a tenant may be a law firm or other organization reviewing emails associated with a third-party entity such as a party to litigation or a third-party entity subject to a subpoena.

[0021] To provide such capabilities, one embodiment of the document analysis system may include a conversation service that organizes emails into conversations so the document analysis system can display these emails in the context of an associated email conversation and allow users to view the conversations and review the emails in the context of these conversations. Specifically, the conversation service may obtain email data collected or obtained from email servers from one or more source systems (e.g., email servers in one or more locations). The conversation service builds a graph of the emails, where the nodes represent data about an email and the edges (connections) in the graph between the nodes of the graph are determined based on metadata associated with the emails or the text content of the emails.

[0022] Moreover, a weighting heuristic may be applied to determine weights associated with edges of the graph, or to identify relationships (e.g., reply relationships) between the emails. Using such a graph, conversations within the email graph may be determined by evaluating the graph to determine conversation trees within the graph. Each conversation tree within the graph representing an email conversation may comprise, for example, a (e.g., disconnected) portion of the graph comprising a set of nodes, including a root node, and a set of edges representing relationships between those nodes. The emails of the conversation represented by the conversation tree may then be evaluated to determine which of those emails are inclusive (or subsumed) and the nodes representing those inclusive (or subsumed) emails marked in the conversation tree. The evaluation of whether one email is subsumed by another email may (or may not) take into account the email content (e.g., body) of the emails themselves, the email metadata of the emails, and the attachments of each email (e.g., an email that includes that content of another email but does not include one or more attachments may, or may not be, considered to subsume that other email).

[0023] Specifically, in some embodiments, nodes of a conversation tree representing an email may include additional information that can be recorded for nodes representing inclusive (or subsumed) emails (or emails which subsume other emails) in the conversation tree. In particular, according to certain embodiments, descendant or parent information can be stored in association with a node representing a subsumed email that indicate which other emails in that conversation subsume the content of that email, or conversely for nodes representing an email which subsumes another email, which other email(s) are subsumed by that email. This subsumed by data can, for example, be an identifier of each email in the conversation (or a node corresponding to that email) that subsumes that email. Thus, each node of the conversation tree representing a subsumed email may identify each email of the conversation that subsumes that email. This information may be stored, for example, as attributes of a node representing an email or other email data associated with the node (e.g., as a "subsumed by" attribute of the node for the email), or may be stored as a relationship in the graph.

[0024] Thus, embodiments may reduce the time required to organize an email dataset into conversations and identify emails with unique (e.g., text) content. Users are able to see how emails relate to each other soon after adding them to the dataset, even when the dataset is large. They are also able to see which emails have unique text content and which emails do not. This allows users

of such document analysis systems to more quickly review of such documents and to better ascertain the emails they may need to review.

[0025] In particular, embodiments may allow users to not only view conversation trees of emails and scope the emails of those conversations tree to inclusive or subsumed emails, but may further allow the scoping of these emails based on other criteria. To illustrate in more detail, in certain cases or with certain document analysis systems it may be difficult to both scope emails (e.g., within a conversation) based on certain criteria where that criteria includes only obtaining (e.g., presenting through the interface) subsumed or inclusive emails (e.g., such that duplicative or redundant emails may be excluded from presentation).

[0026] In fact, in some instances the scoping of emails by one or more criteria (e.g., date range or responsive to a search term query) along with specifying only subsumed or inclusive emails, may actually result in more emails being returned than just scoping the emails of a conversation based on the one or more criteria. From a user's standpoint this may be undesirable. Most users desire to reduce the number of emails obtained in response to such scoping. However, if data indicating only whether an email is (e.g., or is not) subsumed is maintained in association with those emails, an indication that only subsumed emails are desired may cause inclusive or subsumed emails to be found and returned, even if those inclusive or subsumed emails are not responsive (e.g., do not meet) the one or more other scoping criteria. Accordingly, these additional emails may be returned in addition to, or in lieu of, other emails that do fall within the scoping criteria, causing more emails to be returned.

[0027] As discussed, embodiments may address these deficiencies by determining and recording subsumed by data for nodes in a conversation tree representing subsumed emails (or emails which subsume other emails). In particular, according to certain embodiments, descendant or parent information (such as an identifier of an email or node representing that email) can be stored in association with a node representing a subsumed email that indicate which other emails in that conversation subsume the content of that email, or conversely for nodes representing an email which subsumes another email which other email(s) are subsumed by that email.

[0028] Accordingly, when scoping emails of a conversation based on one or more scoping criteria (e.g., other than inclusive or subsumed, such as data range, age, search terms, etc.) and a subsumed or inclusive scoping criteria, the emails of the conversation can be scoped according to the subsumed data associated with the nodes representing those emails in the conversation tree for that conversation. Specifically, a set of emails (e.g., referred to as the relative scope or scoped set) may be determined by traversing the nodes of the conversation tree to determine which emails (e.g., nodes representing those emails) are responsive to the one or more scoping criteria (e.g., are within the data range, age, search terms, etc. defined by that scoping criteria). This scoped set of emails (i.e., emails of the relative scope) can then be evaluated (e.g., iterated through) to determine if any other of the set of emails in the relative scope subsume that email (the email being evaluated) using the subsumed data associated with the nodes representing the emails of the set of emails of the relative scope. If no other email in the set of emails of the relative scope subsumes that email, that email may be added to the final set of emails. This final set of emails responsive to all scoping criteria (e.g., the one or more criteria and the subsumed or inclusive criteria) can then be presented to a user in an interface for the conversation tree.

[0029] In one embodiment, this determination may be done using set operations. For example, the set of identifiers of each of the emails of the relative scope of a conversation may comprise the relative scope set. Each email of the relative scope can then be evaluated to determine if that email (being evaluated) should be added to the final set of emails responsive to all scoping criteria.

[0030] When an email of the relative scope is evaluated to determine if that email should be added to the final set of emails responsive to all scoping criteria, the corresponding set of identifiers of the subsumed by field of the node for that email (if any) may be obtained from the email graph. Each of this set of subsumed by identifiers may thus identify an email of the conversation tree that

subsumes that email (being evaluated). This set of subsumed by identifiers for the email (being evaluated) can be intersected with the relative scope set (comprising identifiers of emails of the relative scope of the conversation). If the intersection of the subsumed by set for the email and the relative scope set is empty (indicating that the email being evaluated is not subsumed by any other email included in the relative scope) the (e.g. identifier of the) email being evaluated may be added to the final set of emails for that conversation while if the intersection of the subsumed by set for the email and the relative scope set is not empty (indicating that the email being evaluated is subsumed by at least one other email included in the relative scope) that (identifier of the email) may not be added to the final set of emails for that conversation.

[0031] In one embodiment, to increase efficiency, especially with respect to the presentation of the emails and related conversations in a user interface, the conversation service may utilize two email graphs, a raw graph comprising nodes representing data on emails from various different sources and a simplified graph derived from the raw graph. The raw graph may also include edges representing relationships between the nodes, with equivalence edges or relationships relating equivalent nodes related to the same email and reply edges or relationships between two nodes, where one node of the two nodes joined by the reply relationship represents email data on a first email that is a reply to a second email represented by the second node joined by the relationship, where the second node represents email data on the second email. Such a raw graph may thus include multiple nodes representing the same email and associated with different types of data which may be obtained from different sources (or from the same source). There may also be a weighting on reply edges denoting, for example, a sum of the signals representing a likelihood of that reply relationship.

[0032] Thus, it will also be noted that because the evaluation of email data to determine reply relationships may be based on various heuristics and there may be a weighting assigned to each reply relationship, in many cases there may be multiple reply relationships in the raw graph associated with a single node, each with an associated weighting. In other words, the raw graph may denote that a particular email (represented by a node) may be a reply to multiple different emails (e.g., represented by distinct nodes), with the weighting on each reply relationship denoting a likelihood or sum of signals for that reply relationship.

[0033] Because of the details which the raw graph represents such a raw graph may prove difficult or inefficient to navigate when manipulating or evaluating such email data for use in providing email analysis or presenting such email data in a user interface (e.g., when real-time or near real-time processing may be desired). Therefore, embodiments of the conversation service may also utilize a simplified graph derived from the raw graph. The simplified graph includes nodes representing distinct emails and reply relationships between these nodes. Specifically, in certain embodiments, the simplified graph may include email nodes that represent equivalent nodes of the raw graph (e.g., nodes of the raw graph which have equivalency relationships between them) collapsed into a single node and reply relationships edges between the email nodes. For the simplified graph, the reply edges may also be collapsed and can then be pruned based on the weighting associated with the resulting reply edge such that each email node has at most one incoming reply edge (e.g., an email can be a reply to only one other email). Using the simplified graph then, conversations may be determined by evaluating the simplified graph to determine conversation trees within the graph. This evaluation may be done, for example, by traversing the simplified graph of trees using a depth first search or other traversal mechanism.

[0034] Again, each conversation tree within the simplified graph representing an email conversation may comprise, for example, a disconnected subgraph of the simplified graph comprising a set of nodes, including a root (or "initiator") node, and a set of edges representing relationships between those nodes. The simplified graph may therefore include a set of subgraphs where each subgraph is a conversation tree representing an email conversation. The emails of the conversation represented by the conversation tree may then be evaluated to determine which of

those emails are inclusive (or subsumed) and the nodes representing those inclusive (or subsumed) emails marked in the conversation tree of the simplified graph.

[0035] Specifically, in some embodiments, for each email not only may a single flag indicating whether that email is subsumed or not but additional information may be recorded for nodes representing inclusive (or subsumed) emails (or emails which subsume other emails) in the conversation tree. In particular, as mentioned descendant or parent information (such as an identifier of an email or node representing that email) can be stored in association with a node representing a subsumed email that indicate which other emails in that conversation subsume the content of that email, or conversely for nodes representing an email which subsumes another email which other email(s) are subsumed by that email. This information may be stored, for example, as attributes of a node representing an email or other email data associated with the node (e.g., date, sender, identifier, etc.) or may actually be stored as a relationship in the graph (e.g., the simplified graph).

[0036] As the simplified graph comprises all the data of a single email (e.g., the reference to the email document, the email metadata, etc.) in a single node, this simplified graph may now be efficiently traversed and used to provide efficient document analysis and review for email conversations to a user of the document analysis system. Specifically, it may be easier to give preference to email document nodes, or determine when an email document for an email has been obtained.

[0037] As discussed, however, in many cases email data may be retrieved or obtained from different email servers (e.g., of different types) at different times. Thus, the conversation trees of the simplified graph determined at one point may be incomplete and new email data regarding such conversation trees may be obtained at (e.g., multiple) subsequent times. Accordingly, such document analysis systems may be required to frequently identify conversation trees pertaining to newly obtained email data and update the identified conversation trees.

[0038] However, because the simplified graph may collapse or eliminate some of the data of the raw graph (e.g., weak reply relationships) a problem may now occur in which the simplified graph does not contain enough data to facilitate the efficient incremental update of the simplified graph when such new email data is obtained. Accordingly, when new email data is received, the raw graph may be utilized to perform an incremental update to the simplified graph.

[0039] According to particular embodiments, then, an incremental raw graph may be determined from the new email data. Specifically, a node may be determined to add to the incremental raw graph based on the identification of a new email from either obtained new email metadata or an obtained new email document. Alternatively, the node may represent newly identified email data (e.g., an email document or email metadata) pertaining to an email already represented by one or more nodes of the raw graph. New edges to add to the raw graph may be determined and added to the incremental raw graph based on the email data associated with the newly determined node. These new edges may include new equivalency edges representing other existing nodes in the raw graph that represent email data for the same email or a reply relationship edge between the new node and a second node of the raw graph representing another email. These new edges and determined second node may be added to the incremental raw graph. Each of the edges of the incremental raw graph can then be followed recursively using the raw graph and each node and relationship transitively connected to the new node in the raw graph may be added to the incremental raw graph.

[0040] Once the incremental raw graph is determined it can be transformed into an incremental simplified graph by collapsing the equivalent nodes of the incremental raw graph to get one single node in the incremental simplified graph representing all equivalent nodes in the incremental raw graph. Additionally, a best reply relationship edge may be selected between any two nodes based on the sum of the weights of all the reply relationship edges between all equivalent nodes for those two nodes (e.g., the nodes collapsed to form those two nodes) in the incremental raw graph. The

determined best reply relationship edges are then added to the incremental simplified graph.

[0041] This incremental simplified graph can then be traversed to determine conversations within the incremental simplified graph and assign identifiers to all the nodes of each conversation tree within the incremental simplified graph. Such a traversal may determine each node of the incremental simplified graph that is a root node (e.g., having no reply relationship edges specifying that node as a reply and representing an originator email) and beginning with each of those nodes traverse the associated conversation tree originating with the node. Such a traversal may, in one embodiment, be done by a recursive common table expression (CTE) in order to efficiently implement such a traversal. A conversation identifier may be, for example, an identifier of the root node (e.g., representing an originator email) of the conversation tree that is propagated to each of the nodes of the conversation tree subgraph during traversal.

[0042] This traversed incremental simplified graph can be used to detect inclusive or subsumed emails in embodiments. The detection of whether an email is subsumed by another email may be based on the text content of the emails and the relationship of the emails in a graph. In some embodiments, this subsumed by determination may also take into account the attachments of each email. For example, one email may be deemed as subsuming another email if they are part of the same conversation, that email is a descendant in the conversation tree for that conversation and includes all the content of the subsumed email (e.g., all the text content without regard to attachments, all the text content and all the corresponding attachments, etc.).

[0043] Specifically, for a conversation tree in the incremental simplified graph each of the branches of the conversation tree may be identified (e.g., where a node of the conversation tree may be associated with one or more branches). The text of the emails of each branch can then be analyzed to determine if the text of any email represented by a node of the branch is duplicated in any descendant of the branch. Based on this determination, the nodes of the branch may be annotated to indicate that the email represented by the node is inclusive or subsumed. As discussed, this subsumed by data or inclusivity data may be stored as attributes of a node representing an email or other email data associated with the node (e.g., date, sender, identifier, etc.) or may actually be stored as a relationship in the graph (e.g., the simplified graph).

[0044] This annotated and traversed incremental simplified graph can then be used in presenting an interface corresponding to any of the conversations trees represented in the incremental simplified graph. Moreover, the incremental raw graph and the incremental simplified graph may then be used to update the raw graph and simplified graph respectively such that both the raw graph and the simplified graph may be incrementally updated based on newly received email data.

[0045] In this manner, then, the document analysis system (and thus users of such a document analysis system) may quickly obtain, analyze, or otherwise utilize, new email data, without the need to reconstruct or traverse the entirety of data structures (e.g., graphs) representing the entirety of the email data obtained from source systems. Instead, such data structures may be efficiently and incrementally updated utilizing incremental data structures representing only portions of the data structures that will need to be altered based on the newly obtained data.

[0046] Moreover, through the use of the encoding of the data structures utilized by embodiments and the data stores or engines used to store and manipulate such data structures, embodiments may provide the advantages that the processing of email data and the manipulation of the representations of email conversations may be efficiently implemented in a distributed or data-local manner using a database engine that may be part of a database platform such as a database as a service. Additionally, in instances where the document analysis system is a multi-tenant platform, the processing may be accomplished as single (or fewer) job executions that can be more efficiently processed for multiple tenants.

[0047] In one embodiment, therefore, email data may include an email conversation comprising a set of emails. One or more scoping criteria (e.g., data ranges, search terms, etc.) can be received and a scoped set of emails determined from the set of emails of the mail conversation based on the

one or more scoping criteria. A final set of emails can be determined based on the scoped set of emails by evaluating the scoped set of emails to eliminate any of the scoped set of emails that are subsumed by another one of the set of scoped emails, and the final set of emails for the email conversation presented to a user.

[0048] In one embodiment, the email data comprises an email graph representing the set of emails and the email graph comprising a set of nodes, wherein each node represents an email of the email conversation and each node comprises a subsumed by field identifying any of the set of emails of the email conversation that include the content of the email represented by that node.

[0049] In some embodiments, the subsumed by field of each node identifies any of the set of emails of the email conversation that include the content of the email represented by that node. This identification may be accomplished, for example, by identifying nodes in the email graph representing any of the set of emails of the email conversation that includes the content of the email represented by that node.

[0050] In one embodiment, system for document analysis can include a data store having an email graph where the document analysis system is adapted to incrementally update the email graph based on email data. The email data can be obtained an incremental graph comprising a set of nodes representing emails, and edges representing relationships between the emails, can be determined. The determination of the incremental graph may include determining a first node for the incremental graph, wherein the first node represents a first email determined from the obtained email data, adding the first node to the incremental graph, and determining one or more edges for the incremental graph. The one or more edges can include at least one edge determined by determining a candidate edge for the incremental graph based on a comparison of first text of the first email to second text of a second email represented by a second node in the email graph or the incremental graph, and adding the candidate edge to the incremental graph as one of the one or more edges.

[0051] Starting with the one or more edges in the email graph, the email graph can be traversed to determine a related set of nodes and a related set of edges of the email graph to add to the incremental graph, where the related set of nodes and a related set of edges of the email graph are directly or indirectly related to the one or more edges. These related set of nodes and related set of edges determined from the email graph can be added to the incremental graph and the incremental graph traversed to identify a set of conversations in the incremental graph, each of the set of conversations comprising a subgraph of the incremental graph. The identified set of conversations can be updated in the email graph based on the incremental graph to incrementally update the email graph based on the obtained email data.

[0052] In some embodiments, the data store comprises a columnar data store. In certain embodiments, the data store is part of a database system comprising a database system including a database engine, wherein the incremental graph is determined at least partially using the database engine.

[0053] In embodiments the email graph comprises a raw graph and a simplified graph, and the incremental graph comprises an incremental raw graph and an incremental simplified graph. In one embodiment, the incremental simplified graph is determined by collapsing the nodes and edges of the incremental raw graph. In a particular embodiment, the incremental simplified graph is traversed to identify the set of conversations in the incremental graph.

[0054] According to one embodiment, the subgraph of each conversation can be traversed to identify each node of the conversation for the subgraph that represents an inclusive email.

[0055] These, and other, aspects of the invention will be better appreciated and understood when considered in conjunction with the following description and the accompanying drawings. The following description, while indicating various embodiments of the invention and numerous specific details thereof, is given by way of illustration and not of limitation. Many substitutions,

modifications, additions, or rearrangements may be made within the scope of the invention, and the invention includes all such substitutions, modifications, additions, or rearrangements.

Description

BRIEF DESCRIPTION OF THE DRAWINGS

[0056] The drawings accompanying and forming part of this specification are included to depict certain aspects of the invention. A clearer impression of the invention, and of the components and operation of systems provided with the invention, will become more readily apparent by referring to the exemplary, and therefore non-limiting, embodiments illustrated in the drawings, wherein identical reference numerals designate the same components. Note that the features illustrated in the drawings are not necessarily drawn to scale.

[0057] FIG. **1** is a block diagram of one embodiment of an architecture including a document analysis system.

[0058] FIG. **2**A is depiction of one embodiment an interface that may be utilized by a document analysis system to present email data.

[0059] FIGS. **2**B and **2**C are examples of an email conversation and the associated scoping of such an email conversation.

[0060] FIGS. **3**A, **3**B and **3**C are a block diagram of example email graphs.

[0061] FIG. **3**D is a depiction of a table schema that may be utilized by embodiment of a document analysis system.

[0062] FIG. **4** is a flow diagram of one embodiment of a method for scoping emails of an email conversation.

[0063] FIG. **5** is a flow diagram of one embodiment of a method for incremental update of a email graph.

[0064] FIG. **6** is a flow diagram of one embodiment of a method for detecting edges for an incremental raw graph.

[0065] FIG. **7** is an example of the detection of a reply edge in an email graph.

[0066] FIG. **8** is a flow diagram of one embodiment of a method for building incremental graphs.

[0067] FIG. **9** is a flow diagram of one embodiment of a method for traversing an incremental graph.

[0068] FIG. **10** is a flow diagram of one embodiment of a method for detecting inclusive emails in an incremental graph.

DETAILED DESCRIPTION

[0069] The disclosure and various features and advantageous details thereof are explained more fully with reference to the exemplary, and therefore non-limiting, embodiments illustrated in the accompanying drawings and detailed in the following description. It should be understood, however, that the detailed description and specific examples, while indicating the preferred embodiments, are given by way of illustration only and not by way of limitation. Descriptions of known programming techniques, computer software, hardware, operating platforms, and protocols may be omitted so as not to unnecessarily obscure the disclosure in detail. Various substitutions, modifications, additions and/or rearrangements within the spirit and/or scope of the underlying inventive concept will become apparent to those skilled in the art from this disclosure.

[0070] Before describing embodiments in more detail, it may be helpful to discuss some context around document analysis systems. As mentioned above, the vast majority of documents that are currently being created, utilized, and maintained are in electronic format. A number of different situations commonly arise that require an analysis or identification of certain relevant electronic documents from a relatively large pool of available electronic documents. For example, in litigation, an entity's documents may need to be reviewed in order to identify documents that may

be relevant to one or more issues in the litigation.

[0071] To illustrate in more detail, parties to litigation typically have to share relevant evidence with opposing counsel through the discovery process. In many cases, each party makes a reasonable search of their records based on some set of terms or keywords and produces the results of the search to the other party. Discovery thus typically involves the gathering of potentially relevant materials, much of it digital, and then reviewing such materials to determine what to be shared with opposite parties. Additionally, during the course of the litigation each party may continually review those documents produced by the opposing party to locate documents relevant to the case at hand. Litigation thus represents a microcosm of a more general problem raised by the high volume of electronic documents present in a variety of contexts. Namely, how can a large volume of electronic documents be understood, reviewed, or searched in order that documents relevant to a particular topic or user's interest may be located.

[0072] To aid users in resolving these problems, a document analysis system, may be provided in a given electronic context. A document analysis system is a computer system used to process a corpus of electronically stored information (referred to as the corpus) and allow users to analyze, review or navigate the information, or search the electronic information to return electronically stored information responsive to a search (also referred to as a query). Items of electronic information that form a corpus may be referred to interchangeably as (electronic) documents, items, files, objects, items, content, etc. and may include objects such as files of almost any type including documents for various editing applications, emails, workflows, etc.

[0073] In particular, emails are ubiquitous within almost all modern enterprises, serving as the almost de-facto form of communication in such enterprises. It is thus often required to obtain and review such emails in the course of a litigation or other type of document review. As such, a document analysis system may provide a mechanism by which users can be presented with these emails and review such emails and associated data. E mails, however, are a problematic subset of the documents analyzed by such a document analysis system as the nature of emails usually results in computationally intensive and slow analysis of such emails.

[0074] Specifically, in many cases it is desirable for document analysis systems to identify and present such emails in the form of "conversation trees", where each conversation tree represents a conversation comprised of a set of associated emails that include an original or root email, (the beginning of the conversation) and all of the subsequent replies and forwards transitively associated with that original email.

[0075] The determination of such email conversations from email data in the document analysis environments is often times complex. Email data may comprise email metadata, or actual emails document themselves, of a large volume of emails (e.g., on the order of hundreds of thousands or millions in some cases). Construction of the conversation trees thus requires the determination of associated emails and the original or root email of such a conversation tree from a large volume of email data. The processing of such email data may thus be computing time and resource intensive.

[0076] Complicating these determinations is the consideration that email data may be retrieved from different email servers (e.g., of different types) at different times. Thus, metadata for emails and the email documents may be obtained at different times. Moreover, the conversation trees determined at one point may be incomplete and new email data regarding such conversation trees may be obtained at (e.g., multiple) subsequent times. Accordingly, such document analysis systems may be required to frequently identify conversation trees pertaining to newly obtained email data and update the identified conversation trees. The incremental identification and update of such conversation trees may be even less performant (e.g., from a computing time and resource standpoint) than the original identification of such conversation trees.

[0077] Other factors also complicate the document analysis of emails in such document analysis systems. Namely, if such emails are obtained from multiple email servers for multiple user accounts of various users in an enterprise, there is likely to be a significant degree of duplicity of

the emails between users. Duplicity may occur, for example, because the same email is sent to a number of recipients at once or because it is often the case that emails quote or include the text of previous emails (e.g., when responding or forwarding an email) and therefore may contain large amounts of duplicative content.

[0078] From a user's perspective it is desirable that such emails including duplicative content are identified by the document analysis system such that the user may review such duplicative content only once. In other words, if an inclusive email (e.g., and any included or subsumed emails) are identified to a user, the user may review all that content in the identified inclusive email and avoid having to review other emails identified as included in an inclusive email. The identification of such inclusive (or subsumed) emails is, however, not a trivial task, being quite computationally expensive, especially when such incremental updates to the email data as discussed above are taken into account.

[0079] It would thus be desirable to have improved systems and methods for document analysis of email data that can efficiently construct and incrementally update representations of email conversations.

[0080] To those ends, among others, attention is directed to embodiments of the document analysis systems disclosed herein. Embodiments of such document analysis systems may obtain email data on emails from source systems and organize and analyze such emails to allow users to review such email using the system. Such a document analysis system may be, for example, a multi-tenant platform that provides document analysis services to users from multiple (e.g., distinct) tenants, including allowing users of these tenants to access and review emails and email data associated with the tenant. For example, in a litigation context such a tenant may be a law firm or other organization reviewing emails associated with a third-party entity such as a party to litigation or a third-party entity subject to a subpoena.

[0081] To provide such capabilities, the document analysis system may include a conversation service that organizes emails into conversations so the document analysis system can display these emails in the context of an associated email conversation and allow users to view the conversations and review the emails in the context of these conversations. Specifically, the conversation service may obtain email data collected or obtained from email servers from one or more source systems (e.g., email servers in one or more locations). The conversation service builds a graph of the emails, where the nodes represent data about an email and the edges (connections) in the graph between the nodes of the graph are determined based on metadata associated with the emails or the text content of the emails. These email graphs may be efficiently updated as new email data is obtained such that the document analysis system may quickly organize emails into conversations for utilization by users in reviewing these emails in context.

[0082] Looking now at FIG. **1**, a block diagram of one embodiment of a document analysis system employing hierarchical clustering of document portions is depicted. The document analysis system **101** is part of computing environment **100** including a database system **105**, document analysis system **101**, and one or more client computers **130**.

[0083] In the depicted embodiment document analysis system **101** may include one or more (virtual or physical) servers or other type of computing device utilizing a central processing unit **112** connected to a memory and a data store **118** (e.g., via a bus). Central processing unit **112** may represent a single processor, multiple processors, a processor(s) with multiple processing cores and the like. Data store **118** may include a volatile or non-volatile non-transitory storage medium such as RAM, hard disk drives, flash memory devices, optical media, or the like. Document analysis system **101** may be connected to a data communications network such as the Internet, a local area network (LAN), a wide area network (WAN), a cellular network or some other network or combination of networks.

[0084] Data store **118** stores computer executable instructions **119**. Computer executable instructions **119** can represent one or more multiple programs or operating system instructions. In

one embodiment, instructions **119** are executable to provide document analysis application **122**. Document analysis application **122** may be implemented on the same computing systems or can be distributed across multiple computing systems, platforms or physical or virtual server. The document analysis system **101** can be a multi-tenant document analysis system whereby the services provided by the document analysis system **101** may be accessed by users associated with different entities (e.g., enterprises, organizations, etc.) to perform those services on data associated with those entities. Again, it will be noted here that while embodiments described and depicted with respect to FIG. **1** include a deployment of a document analysis system on a physical computing device other embodiments may include the document analysis system deployed as a service on, for example, a cloud computing environment or otherwise deployed without loss of generality.

[0085] Client computer system **130** may include components similar to those of the server of document analysis system **101**, such as CPU **138** and data store **140**. Additionally, client computer system **130** may include executable instructions **132** to provide user interface **134** that allows a user to interact with document analysis system **101** to review the documents **107** of the corpus. These instructions **132** may have, for example, been provided by document analysis system **101** in response to an access by client computer **130**. For example, user interface **134** may be provided through a web browser, file system interface or other method without loss of generality. Moreover, user interface **134** may interact (e.g., issue requests, receive responses, etc.) with a corresponding interface **136** of the document analysis application **122**, which may be, for example a web services interface, an Application Programming Interface (API) or another type of interface.

[0086] Again, those skilled in the art will appreciate that document analysis system **101** shown in FIG. **1** is merely an example of a computing system and embodiments of a document analysis system that may be implemented using other computing systems (e.g., desktop computers, laptops, mobile computing devices, services platforms, cloud computing platforms or other computing devices or platforms with adequate processing and memory) including multiple computers acting together to provide a document analysis system (e.g., a cluster of servers or other computing devices connected by a network acting together to provide the document analysis system). Similarly, client computer **130** may include any suitable desktop computer, laptop, mobile device, server, or other computing system.

[0087] Database system **105** utilized by document analysis system **101** may comprise a file server or database system or other storage mechanism remotely or locally accessible by document analysis system **101** which, according to an embodiment, may be almost any SQL or NoSQL platform. Specifically, in one embodiment database system **105** may be, for example, a columnar data store such as Hbase, Bigtable, or that provided by Snowflake. Additionally, in some embodiments database system **105** may include database engine **109**. Such a database engine **109** may be used for processing or otherwise manipulating or performing operations on data stored in database system **105** in a data local manner (e.g., performing operations close to the location of the data which is being operated on). In one embodiment, the database engine **109** may include multiple compute nodes or clusters that can perform parallel processing on data stored at the database system **105** based on requests or instructions received at database engine **109**, where that data may be stored in internally optimized compressed columnar format in a cloud based storage infrastructure.

[0088] Thus, at some time interval, document analysis system **101** may obtain email data from one or more source systems and store the email data **107** at the database system **105**. These source systems may, for example, be email servers, user's computers, or other sources within an enterprise environment **111**, or another type of source system that may be external to an enterprise environment **111**. This email data **107** may include for example, the email documents or email metadata associated with emails sent from or received by particular users or email addresses. In some embodiments, the email documents received in accordance with such email data **107** may be

assigned an identifier and may be further evaluated, processed, parsed, etc. to separate the email document into a set of text segments or other data **121** that may individually be stored and accessible at the database system **105** in association with the email data **107** and the corresponding email document. In one embodiment, these text segments **121** may be parsed out from an email or separated based on a portion of the email in which the text appears, or whether the text is original or quoted text within an email document. Thus, a text segment **121** may represent headers or send dates of an email, a text segment **121** may represent (e.g., a normalized) subject, a text segment **121** may represent original text of an email, a text segment **121** may represent a text of a first level of quotation, a text segment **121** may represent a second level of quoted text in an email, etc.

[0089] As discussed, associated email data **107** may be obtained at different points in time, where this email data **107** may be different types of email data. For example, in a litigation context, email data may be produced for one or more users or email addresses at a first point in time and email data for other users or email addressed may be produced at a subsequent point in time. Additionally, even email data for the same user or email address may be produced at different times, for example, email metadata for emails of a user may be obtained from one source system at a first time while the actual emails (e.g., email documents) may be obtained from another (or the same) source system at a second subsequent point in time. Additionally, as has been discussed, document analysis system **101** may be a multi-tenant platform, thus the email data **107** obtained at a given point may be associated with a particular tenant of the document analysis system **101** and stored in association with that tenant at the database system **109**. Accordingly, in one embodiment, the email data **107** obtained by the document analysis system **101** may include email data **107** obtained at different times for different tenants.

[0090] The document analysis application **122** may provide an interface (e.g., interface **136**) and associated service through which users may review this email data **107** (e.g., email data associated with a tenant with which the user is associated). In many cases, such an interface may present an interface by which these emails may be accessed an reviewed in the form of "conversation trees", where each conversation tree represents a conversation comprised of a set of associated emails that include an original or root email, (the beginning of the conversation) and all of the subsequent replies and forwards transitively associated with that original email. Such an interface may allow a user to provide one or more scoping criteria (e.g., such as data range, age, search terms, etc.) so emails presented through the interface may be scoped (e.g., the emails of a presented conversation can be defined by the scoping criteria).

[0091] Referring briefly to FIG. **2**A, one embodiment of an interface that may be utilized by a document analysis system is presented. As can be seen here, the interface **200** may include an area where a conversation tree **202** comprising a graphical depiction of a related set of emails may be presented. The conversation tree **202** thus graphically depicts each of the emails of a conversation tree **202** that have been determined from email data and the relationships between them (e.g., which email is a root email, which emails are responsive to other emails, etc.). Additionally, the conversation tree **202** may graphically depict which emails are inclusive of other emails or are subsumed in other emails. The interface **200** may also include an area **204** where the email document associated with a selected email (e.g., as selected from the conversation tree **202**) may be displayed (if available at the document analysis system) and an area **206** where metadata about the selected email may be displayed. The presented conversation tree **202** may thus include the minimum set of emails of the conversation tree **202** that allows a user to review the complete contents of all emails of the conversation represented by that conversation tree **202**. In particular, emails of the conversation that are subsumed by another email of the conversation tree **202** may be removed from those emails presented in the conversation tree.

[0092] As may be imagined, when a user provides one or more scoping criteria including an inclusive or subsumed scoping criteria for emails (e.g., email conversations), it may be desirable that an interface for presentation of scoped email conversations (such as that presented in FIG. **2**A)

present these email conversations in a manner that makes intuitive sense to a user. In no small part this may mean including only emails from a conversation that meet the provided scoping criteria. For ease of reference, the set of emails of a conversation that meet the scoping criteria may be referred to as the relative scoped set of emails (of the conversation).

[0093] Additionally, however, in order to ease their review process, to speed the search or access of emails and thus the generation of the interface, to reduce computer resource usage, or for other reasons, a user may also desire that emails that are subsumed by other emails of the relative scoped set of email not be presented in the set of emails of the conversation that are presented to the user (e.g., in the interface). In other words, a user may desire that the presentation of the set of relative scoped emails of a conversation comprise a minimal set of these relative scoped emails (e.g., that a user may need to review) while still encompassing all the content (e.g., email text or attachments) of the relative scoped set. Thus, a user may specify a scoping criteria defining an inclusive or subsumed criteria to indicate that only subsumed or inclusive emails are to be included in the emails of the conversation presented.

[0094] It may, however, be difficult to both determine a set of relative scoped emails of a conversation that meet certain scoping criteria (e.g., other than inclusive or subsumed, such as data range, age, search terms, etc.) and to determine which of that set of relative scoped emails of a conversation are inclusive of (or subsumed by) other emails within that set of relative scoped emails. Moreover, it may especially difficult to do so in a dynamic manner as the emails of the relative scope (e.g., responsive to the scoping criteria) are determined and presented to a user. This is especially true given the somewhat arbitrary nature of the scoping criteria that a user may provide such as search criteria, etc.

[0095] A simple example will serve to illustrate. Looking then at FIG. **2**B, suppose an email conversation **210** comprises six emails **220**. The first email **220***a* was sent on Oct. 29, 2021, and included two attachments **222***a*, **222***b*. The second email **220***b* sent on Oct. 30, 2021 (e.g., which may have been a forward or reply, etc.) included the text content of the first email **220***a* and also included the same two attachments **222***a*, **222***b*. The third email **220***c* was sent on Oct. 30, 2021, and included the text content of both the first email **220***a* and the second email **220***b*. The fourth email **220***d* was sent on Oct. 31, 2021, and included the text content of the first email **220***a*, the second email **220***b* and the third email **220***c*. The fifth email **220***e* was sent on Nov. 2, 2021, and included the text content of the first email **220***a*, the second email **220***b*, the third email **220***c* and the fourth email **220***d*. The sixth email **220***f* was also sent on Nov. 2, 2021, and included the text content of the first email **220***a*, the second email **220***b*, the third email **220***c*, the fourth email **220***d* and the fifth email **220***e* along with a different attachment **222***c*.

[0096] Now, suppose the user specifies scoping criteria **224** comprising a text term "pumpkin" and a date range of Oct. 1, 2021 to Oct. 31, 2021. Thus, the relative scope of the email conversation should comprise any emails **220** (e.g., including attachments **222**) that include the term "pumpkin" and were sent (e.g., or existing, received, etc.) on or between Oct. 1, 2021 to Oct. 31, 2021. Suppose further, that attachment **222***a*, the text content of the third email **220***c*, the text content of the fourth email **220***d* all include the term "pumpkin". Thus, first email **220***a*, second email **220***b*, third email **220***c* and fourth email **220***d* may be deemed part of the relative scope **226** of the email conversation determined based on scoping criteria **224** as these emails **220** include both the term "pumpkin" and fall within the data range of the scoping criteria **224**. While fifth and sixth email **220***e*, **220***f* may include the term "pumpkin", they fall outside the data range specified in scoping criteria **224** and are thus not in the relative scope of email conversation **210** as determined using scoping criteria **224**.

[0097] Moving now to FIG. **2**C, if the user has specified that subsumed emails are not be included in emails returned for a conversation, what would be desired for email conversation **210** and scoping creature **224** would be to return second email **220***b* (including attachments **222***a*, **222***b*) and fourth email **220***d*, as these two emails **220***b*, **220***d* encompass are the minimum set of emails that

encompass all the content (e.g., text and attachments **222**) included in the relative scope **226** determined based on scoping criteria **224**. Such behavior is, however, not straightforward. In fact, in many cases, emails outside a relative scope may be determined and returned when scoping criteria are specified along with a specification that only subsumed or inclusive emails are to be included (e.g., in the example, referring to FIG. **2**C again, sixth email **220***f* may be determined and returned even though it is not in relative scope **226**. Thus, certain behaviors of document analysis systems with respect to presentation of email conversations may not be inline with a user's intuitive sense as emails that do not actually meet the scoping parameters provided by a user may actually be presented to the user when presenting an email conversation.

[0098] Accordingly, when scoping emails of a conversation based on one or more scoping criteria (e.g., other than inclusive or subsumed, such as data range, age, search terms, etc.) and a subsumed or inclusive scoping criteria, the emails of the conversation can be scoped according to the subsumed data associated with the nodes representing those emails in the conversation tree for that conversation.

[0099] Returning to FIG. **1** now, embodiments of document analysis system **101** may address these deficiencies by determining and recording subsumed by data for nodes in a conversation tree representing subsumed emails (or emails which subsume other emails). In particular, according to certain embodiments, descendant or parent information (such as an identifier of an email or node representing that email) can be stored in association with a node representing a subsumed email that indicate which other emails in that conversation subsume the content of that email, or conversely for nodes representing an email which subsumes another email which other email(s) are subsumed by that email.

[0100] Accordingly, when scoping emails of a conversation based on one or more scoping criteria (e.g., other than inclusive or subsumed, such as data range, age, search terms, etc.) and a subsumed or inclusive scoping criteria, the emails of the conversation can be scoped according to the subsumed data associated with the nodes representing those emails in the conversation tree for that conversation [0101] in order to provide such capabilities, the document analysis system **101** may include a conversation service **124** that organizes emails into conversations so the document analysis system can display these emails in the context of an associated email conversation and allow users to view the conversations and review the emails in the context of these conversation. Specifically, the conversation service **124** may obtain email data **107** collected or obtained from the one or more source systems (e.g., email servers in one or more locations).

[0102] The conversation service **124** builds graph of the emails (an email graph), where the nodes of the graph represent emails and data about that email and the edges (connections) in the graph between the nodes of the graph are determined based on metadata associated with the emails or the textual content of the email.

[0103] Moreover, conversation service **124** may apply a weighting heuristic to determine weights associated with edges of the graph, or to identify relationships (e.g., reply relationships) between the emails. Using such a graph, conversations within the email graph may be determined by evaluating the graph to determine conversation trees within the graph. Each conversation tree within the graph representing an email conversation may comprise, for example, a disconnected portion of the graph comprising a set of nodes, including a root node, and a set of edges representing relationships between those nodes. The emails of the conversation represented by the conversation tree may then be evaluated to determine which of those emails are inclusive (or subsumed) and the nodes representing those inclusive (or subsumed) emails marked in the conversation tree.

[0104] Specifically, embodiments may maintain not only may a single flag indicating whether that email is subsumed or not, but additionally recording information for nodes representing inclusive (or subsumed) emails (or emails which subsume other emails) in the conversation tree. In particular, according to certain embodiments, descendant or parent information can be stored in

association with a node representing a subsumed email that indicate which other emails in that conversation subsume the content of that email, or conversely for nodes representing an email which subsumes another email which other email(s) are subsumed by that email.

[0105] In particular, in one embodiment, to increase efficiency, especially with respect to the presentation of the emails and related conversations in a user interface as discussed, the conversation service **124** may utilize an email graph **172**. In one particular embodiment, this email graph **172** may comprise two email graphs, a raw graph **113** comprising nodes representing data on emails as determined from various different sources, and a simplified graph **115** derived from the raw graph **113**. The raw graph **113** may also include edges representing relationships between the nodes, with equivalence edges or relationships relating equivalent nodes related to the same email and reply edges or relationships between two nodes, where one node of the two nodes joined by the reply relationship represents email data on a first email that is a reply to a second email represented by the second node joined by the relationship, where the second node represents email data on the second email. The presence of the emails and the reply relationships can be derived, for example, from the email documents or the metadata associated with such documents, such as the SMTP metadata or MAPI metadata. Such a raw graph **113** may thus include multiple nodes representing the same email and associated with different types of data which may be obtained from different sources (or from the same source). There may also be a weighting on reply edges denoting, for example, a sum of the signals representing a likelihood of that reply relationship.

[0106] Thus, it will also be noted that because the evaluation of email data to determine reply relationships may be based on various heuristics, and there may be a weighting assigned to each reply relationship, in many cases there may be multiple reply relationships in the raw graph associated with a single node, each with an associated weighting. In other words, the raw graph may denote that a particular email (represented by a node) may be a reply to multiple different emails (e.g., represented by distinct nodes), with the weighting on each reply relationships denoting a likelihood or sum of signals for that reply relationship.

[0107] It may be helpful here to give an example of a portion of a raw graph such that the nodes and edges comprising raw graph **113** may be better understood. FIG. **3**A depicts just such an example of a portion of a raw graph. Here, the raw graph **302** may include a set of email nodes **304** representing data of an email. Thus, nodes **304** may represent a piece of metadata about an email or may represent an email document of an email. Thus, for example, nodes, **304***a*, **304***b* and **304***c* may be associated with document identifiers (e.g., "Docs/4, "Docs/2" and "Docs/3") where those document identifiers may be identifiers of documents (e.g., the actual sent email, attachments, etc.) for an email that are stored at the document analysis system (e.g., "Doc/4" may be an identifier of an email document for an email. Nodes **304***d*, **304***e*, **304***f*, **304***g*, **304***h* and **304***i* may accordingly be associated with identifiers for an email as obtained from email metadata obtained from a source system.

[0108] As an example, nodes **304***d*, **304***e*, **304***f*, **304***g* may each represent emails determined from Simple Mail Transfer Protocol (S MTP) metadata and represent those emails with identifiers determined from this email metadata (e.g., "<MSG-0>"), while nodes **304***h* and **304***i* may each represent emails determined from Messaging Application Programming Interface (MAPI) metadata and represent those emails with identifiers determined from this email metadata (e.g., "1234AB").

[0109] As has been noted, emails determined from different sources may in actuality be the same email. This may occur, for example, when the same email is independently identified by the conversation service in multiple sources and represented by an equivalent node with the identifier for that email associated with each source. The nodes **304** representing emails determined from various sources may thus have equivalence relationships **306** amongst themselves denoting that the email represented by the related nodes is the same email. For example, equivalence relationship **306***a* between node **304***a* and **304***d* may represent that the email that included the document identified in the document analysis system by the identifier "Doc/4" and represented by node **304***a*

is the same email as the email (e.g., independently) identified by "<MSG-0>" in SMTP email metadata and represented by node **304***d*. Note here that, in some cases, emails may only be identified from certain sources, or at certain points, data identifying particular emails may have only been received from certain source system. Thus, for example, a node **304***e* may represent an email that has been determined to exist using SMTP metadata and is identified my "<MSG-1>", but that no corresponding document for that email has been identified or received and that email has not been identified in data from any other source system (e.g., no other metadata on that email has been identified).

[0110] The raw graph **302** may also include reply relationships **308** between nodes **304** denoting that an email represented by a node **304** is a reply to another email represented by the other node **304** joined by the relationship **308**. These reply relationships may be directional such that they indicate which of the related nodes represents the email that is the reply and which of the related nodes represents the email that was replied to. For example, relationship **308***a* may denote that the email represented by node **304***d* (e.g., identified by "<MSG-0>" in SMTP metadata) is a reply to an email represented by node **304***e* (e.g., identified by "<MSG-a>" in SMTP metadata). Each reply relationship **308** may also have an associated weight determined by a weighting heuristic, where the weight is numerical representation of the strength of the relationship (e.g., an indication of how likely the reply relationship between the emails represented by the nodes is).

[0111] It will be noted here that, in some cases, reply relationships **308** in the raw graph **302** may only be between nodes **304** derived from the same source (or type of source) or representing the same type of data. This may be the case, for example, because these reply relationships may be derived or determined based on the data derived from the source system. Thus, reply relationships **308** may only be, for example, between nodes **304** that represent emails as determined from email documents (e.g., nodes **304***a*, **304***b*, **304***c*), between nodes **304** that represent emails as determined from SMTP metadata (e.g. nodes **304***d*, **304***e*, **304***f, ***304***g*), or between nodes **304** that represent emails as determined from MAPI metadata (e.g. nodes **304***h, ***304***i*). In certain embodiments then, the weighting heuristic utilized in determining a weight for a reply relationship **308** may be based on the source system (or data determined therefrom) from which the related nodes **304** were derived. The weighting heuristic may be, for example, a numerical representation of the signals derived from a source system that were evaluated to determine such a reply relationship **308** should exist.

[0112] Referring back to FIG. **1**, because of the details which the raw graph **113** represents, such a raw graph **113** may prove difficult or inefficient to navigate when manipulating or evaluating such email data for use in providing email analysis or presenting such email data in a user interface (e.g., when real-time or near real-time processing may be desired). For instance, it may be desired to give priority to the nodes of the email graph **172** that are associated with email documents when traversing an email graph **172** in order to facility presentation of those documents to a user in an interface of the document analysis system **101**. Because of the nature of graphs, among other reasons, it may be difficult to give preference to the email document nodes when traversing raw graph **113**.

[0113] As such, in order to be able to give priority to nodes of an email graph that are associated with documents of an email when traversing an email graph embodiments of the conversation service **124** may also create and utilize a simplified graph **115** derived from the raw graph **113**. The simplified graph **115** includes nodes representing distinct emails and reply relationships between these notes. Specifically, in certain embodiments, the simplified graph **115** may include email nodes that represent equivalent nodes of the raw graph **113** (e.g., nodes of the raw graph which have equivalency relationships between them) collapsed into a single node and reply relationships edges between the email nodes. For the simplified graph **115**, the reply edges of the raw graph **113** may also be collapsed and can then be pruned based on the weighting associated with the resulting reply edge such that each email node of the simplified graph **115** has at most one incoming reply

edge (e.g., an email can be a reply to only one other email). As the simplified graph **115** comprises all the email data (or reference to such data) of a single email (e.g., the reference to the email document, the email metadata, attachments, etc.) in a single node, this simplified graph may now be efficiently traversed and used to provide efficient document analysis and review for email conversations to a user of the document analysis system. For example, it may be easier to give preference to, or determine when, an email document for an email has been obtained and give priority to such email documents during traversal.

[0114] Again, it may be helpful here to give an example of a portion of a simplified graph such that the nodes and edges comprising simple graph **115** may be better understood. FIG. **3**B depicts an example of a simplified graph corresponding to the example raw graph of FIG. **3**A. Here email nodes **304** (e.g., **304***j*, **304***k*, **304***l*, **304***m*) represent emails. These email nodes **304** are determined based on email nodes in the corresponding raw graph **302** that represent the same email as determined from the equivalency relationships **306** of the raw graph **302**. In particular, each email node **304** of the simplified graph **320** is a single node representing the email and associated with all the data of every equivalent node **304** in the raw graph **302**. For example, node **304***l* of the simplified graph **320** may represent the email represented by nodes **304***b*, **304***f* and **304***h* in the corresponding raw graph **302** and be associated with all the identifiers for that email as determined from those collapsed nodes (e.g., "Docs/2", <MSG-2>" and "1234AB").

[0115] Similarly, each reply relationship **308** of the simplified graph **320** between two nodes **304** may represent that an email represented by a node **304** is a reply to another email represented by the other node **304** joined by the relationship **308**. These reply relationships may be directional such that they indicate which of the related nodes **304** represents the email that is the reply and which of the related nodes represents the email that was replied to. For example, relationship **308***b* may denote that the email represented by node **304***l* is a reply to an email represented by node **304***m*. Node **304***m* however, may have no outgoing reply relationships. In other words, the email represented by node **304***m* may not be a reply to any other email, thus the email represented by node **304***m* may be considered the initiator mail and node **304***m* may be considered the "root" of the conversation tree. These reply relationships **308** between nodes **304** in the simplified graph **320** may be determined based on corresponding reply relationships **308** of the raw graph **302** between equivalent nodes **304** of the raw graph **302** representing the same emails. Specifically, reply relationships between equivalent nodes **304** in the raw graph **302** may be collapsed into a single reply relationship **308** in the simple graph **320** having a weight determined based on the collapsed reply relationships of the raw graph **302**.

[0116] To illustrate by way of example, here node **304***l* in the simplified graph **320** may represent an email represented by each of equivalent nodes **304***b*, **304***f* and **304***h* in the raw graph **302**, while node **304***m* in the simplified graph **320** may represent an email represented by each of equivalent nodes **304***c*, **304***g* and **304***i* of the raw graph **302**. Relationship **308***c* exists between nodes **304***f* and **304***g* and has a weight of 0.9, relationship **308***d* exists between nodes **304***b* and **304***c* and has a weight of 1.1 and relationship **308***e* exists between nodes **304***h* and **304***i* and has a weight of 1.0. Relationship **308***b* between nodes **304***l* and **304***m* in the simplified graph **320** may thus be determined by collapsing relationships **308***c*, **308***d* and **308***e* of the raw graph **302** into a single relationship having a weight of 3.0 (e.g., the sum of the individual weights of the collapsed relationships **308***c*, **308***d* and **308***e*).

[0117] It will be noted that in the simplified graph **320** it may be desired that each email node **304** of the simplified graph **115** has at most one incoming reply edge (e.g., an email can be a reply to only one other email). Thus, if there are multiple possible incoming reply relationships to a node **304** as determined from the raw graph **302**, the reply relationship **308** with the highest weight (e.g., as determined from combined weights of the corresponding collapsed relationships in the raw graph) may be selected and included in the simplified graph **320** while other determined possible reply relationships for the node may be pruned (e.g., not represented in the simplified graph).

[0118] Retuning to FIG. **1**, the conversation service **124** can thus determine and present conversations by evaluating the email graph **172** to determine conversation trees within the email graph **172**. In one embodiment, for example, simplified graph **115** may be evaluated to determine conversation trees in simplified graph **115**. This evaluation may be done, for example, by traversing the simplified graph **115** for trees using a depth first search or other traversal mechanism. Each conversation tree within the simplified graph representing an email conversation may comprise, for example, a disconnected subgraph of the simplified graph comprising a set of nodes, including a root (or "initiator") node, and a set of edges representing relationships between those nodes. The simplified graph **115** may therefore include a set of subgraphs where each subgraph is a conversation tree representing an email conversation. For instance, the simplified graph **320** depicted in FIG. **3**B is an example conversation tree that may be a subgraph (portion of) a larger simplified graph.

[0119] Conversation service **124** may evaluate the emails of the conversation represented by each conversation tree (e.g., conversation tree **320**) to determine which of those emails are inclusive (or subsumed) and the nodes representing those inclusive (or subsumed) emails marked in the conversation tree of the simplified graph. The evaluation of whether one email is subsumed by another email may (or may not) take into account the email content (e.g., body) of the emails themselves, the email metadata of the emails, and the attachments of each email (e.g., an email that includes that content of another email but does not include one or more attachments may, or may not be, considered to subsume that other email).

[0120] Accordingly, nodes of a conversation tree representing emails of that conversation may include additional information that can be recorded for nodes representing inclusive (or subsumed) emails (or emails which subsume other emails) in the conversation tree. In particular, according to certain embodiments, descendant or parent information can be stored in association with a node representing a subsumed email that indicates which other emails in that conversation subsume the content of that email, or conversely for nodes representing an email which subsumes another email, which other email(s) are subsumed by that email. This subsumed by data can, for example, be an identifier of each email in the conversation (or a node corresponding to that email) that subsumes that email. Thus, each node of the conversation tree representing a subsumed email may identify each email of the conversation that subsumes that email. This information may be stored, for example, as attributes of a node representing an email or other email data associated with the node (e.g., as a "subsumed by" attribute of the node for the email), or may be stored as a relationship in the graph.

[0121] As shown in the diagram of FIG. **3**C, in one embodiment an email graph **372** may thus comprise a set of conversation trees **380** where each of the conversation trees **380** may include nodes representing the emails of the conversation associated with the conversation tree wherein information can be stored in association with a node representing a subsumed email that indicates which other emails in that conversation subsume the content of that email. For example, suppose that conversation tree **380***a* represents email conversation **210** as depicted in FIGS. **2**B and **2**C. In this example, each node **374** of the conversation tree **380***a* can represent an email **220** of conversation **210**.

[0122] In particular, node **374***f* represents sixth email **220***f*, node **374***e* represents fifth email **220***e*, node **374***d* represents fourth email **220***d*, node **374***c* represents third email **220***c*, node **374***b* represents second email **220***b*, and node **374***a* represents first email **220***a*. Notice here that node **374***a* for first email **220***a* may be associated with document identifiers identifying a document (e.g., Doc/1) corresponding to the email data (e.g., text of the email) and also document identifiers identifying attachments **222***a*, **222***b* to that email **220***a* (e.g., Doc/2 and Doc/3). Similarly, node **374***b* for second email **220***b* may be associated with document identifiers identifying a document (e.g., Doc/4) corresponding to the email data (e.g., text of the email), and also document identifiers identifying attachments **222***a*, **222***b* to that email **220***b* (e.g., Doc/2 and Doc/3). Node **374***f* for sixth

email **220f** may thus be associated with document identifiers identifying a document (e.g., Doc/8) corresponding to the email data (e.g., text) and also document identifiers identifying attachment **222c** to that email **220f** (e.g., Doc/9).

[0123] Each of the nodes **374** of the conversation tree **380a** may also be associated with a "subsumed by" field, attribute, or property (e.g., used here interchangeably to refer to associated data without loss of generality) that identifies any other emails in the conversation tree **270a** that subsume the content of that email. These identifiers may, for example, be identifiers of nodes **374** representing the conversation tree (e.g., an identifier associated with the node), an identifier of one or more documents, or some other type of identifier associated with an email. Accordingly, in this example, node **374e** representing fifth email **220e** may include an identifier (e.g., "6") in a subsumed by field that indicates that the email represented by node **374e** (which is associated with the Node ID "5") is subsumed by the (e.g., sixth) email **220f** represented by node **374f** (which is associated with the Node ID "6"). Node **374d** representing fourth email **220d** may include identifiers (e.g., "5" and "6") in a subsumed by field that indicates that the email represented by node **374d** (which is associated with the Node ID "4") is subsumed by both the (e.g., fifth) email **220e** represented by node **374e** (which is associated with the Node ID "5") and the (e.g., sixth) email **220f** represented by node **374f** (which is associated with the Node ID "6"). Node **374c** representing fourth email **220d** may include identifiers (e.g., "4", "5", and "6") in a subsumed by field that indicates that the email represented by node **374c** (which is associated with the Node ID "3") is subsumed by the (e.g., fourth) email **220d** represented by node **374d** (which is associated with the Node ID "4"), the (e.g., fifth) email **220e** represented by node **374e** (which is associated with the Node ID "5"), and the (e.g., sixth) email **220f** represented by node **374f** (which is associated with the Node ID "6"). Node **374a** representing first email **220a** may include an identifier (e.g., "2") in a subsumed by field that indicates that the email represented by node **374a** (which is associated with the Node ID "1") is subsumed by the (e.g., second) email **220b** represented by node **374b** (which is associated with Node ID "2").

[0124] To understand how such nodes may be maintained in a document analysis system **101**, one embodiment of a table structure that may be utilized for storing email graphs in a columnar data store is depicted in FIG. **3D**. Embodiments of such a table structure may be a functionally recursive manner of encoding a graph structure that matches well with the capabilities of a columnar data store, including, for example, common table expressions or recursive common table expressions. In this example, the matter identifier (e.g., the "matter_uuid") may correspond to an identifier associated with a tenant of the document analysis system, while the "conv_id" may correspond to an identifier for a conversation within a graph (e.g., an identified conversation tree). An instance (e.g., column or row) of a "final_graph" table may correspond to an email node while an instance (e.g., column or row) of a "edges" may correspond to a reply relationship.

[0125] Referring back to FIG. **1**, the subsumed by information included in nodes of the conversation trees of email graph **172** may be used to determine emails to present in an interface (e.g., interface **136**) when a user is accessing or reviewing such conversation trees. For example, an interface of document analysis system **101** (e.g., interface **136**) may allow a user to provide one or more scoping criteria (e.g., such as data range, age, search terms, etc.) along with a subsumed or inclusive scoping criteria indicating that subsumed by emails should not be included in the set of emails of a conversation tree presented to the user. Conversation service **124** may be provided with this scoping criteria (e.g., including the subsumed or inclusive scoping criteria). Conversation service **124** can then determine conversation trees of email graph **172** that are responsive to the scoping criteria (e.g., where at least one email or attachment associated with that conversation tree meets the scoping criteria). For each conversation tree that meets the scoping criteria, the emails of the corresponding conversation may be scoped according to the subsumed data associated with the nodes representing those emails in the conversation tree for that conversation (e.g., the subsumed by data associated with nodes of the conversation tree) to determine a final set of emails for that

conversation (e.g., conversation tree). This final set of emails for each conversation tree can then be made available, or presented, to the user.

[0126] Specifically, to scope a conversation (e.g., a conversation tree) based on scoping criteria (e.g., other than the subsumed or inclusive scoping criteria), a set of emails (e.g., referred to as the relative scope or scoped set) may be determined by traversing the nodes of the conversation tree to determine which emails (e.g., nodes representing those emails) are responsive to the one or more scoping criteria (e.g., are within the data range, age, search terms, etc. defined by that scoping criteria). This scoped set of emails (i.e., emails of the relative scope) can then be evaluated (e.g., iterated through) to determine if any other of the set of emails in the relative scope subsume that email (the email being evaluated) using the subsumed by data associated with the nodes representing the emails of the set of emails of the relative scope. If no other email in the set of emails of the relative scope subsumes that email, that email may be added to the final set of emails. This final set of emails responsive to all scoping criteria (e.g., the one or more criteria and the subsumed or inclusive criteria) can then be presented to a user in an interface for the conversation tree.

[0127] In one embodiment, this determination may be done using set operations. For example, the set of identifiers of each of the emails of the relative scope of a conversation may comprise the relative scope set. Each email of the relative scope can then be evaluated to determine if that email (being evaluated) should be added to the final set of emails responsive to all scoping criteria. When an email of the relative scope is evaluated to determine if that email should be added to the final set of emails responsive to all scoping criteria, the corresponding set of identifiers of the subsumed by field of the node for that email (if any) may be obtained from the email graph. Each of this set of subsumed by identifiers may thus identify an email of the conversation tree that subsumes that email (being evaluated). This set of subsumed by identifiers for the email (being evaluated) can be intersected with the relative scope set (comprising identifiers of emails of the relative scope of the conversation). If the intersection of the subsumed by set for the email and the relative scope set is empty (indicating that the email being evaluated is not subsumed by any other email included in the relative scope) the (e.g. identifier of the) email being evaluated may be added to the final set of emails for that conversation while if the intersection of the subsumed by set for the email and the relative scope set is not empty (indicating that the email being evaluated is subsumed by at least one other email included in the relative scope) that (identifier of the email) may not be added to the final set of emails for that conversation.

[0128] FIG. **4** depicts one embodiment of method for determining a scoped set of emails for a conversation that accounts for inclusive or subsumed emails. Initially, a conversation tree for the conversation may be obtained (STEP **402**). The emails comprising a relative scope set for that conversation can then be determined based on a set of scoping criteria (STEP **404**) (e.g., other than a subsumed or inclusive scoping criteria). This relative scope set may be determined by traversing the nodes of the conversation tree to determine which emails (e.g., nodes representing those emails) are responsive to the one or more scoping criteria (e.g., are within the data range, age, search terms, etc. defined by that scoping criteria). The relative scope set can thus comprise the set of identifiers for emails that are associated with (e.g. responsive to) that scoping criteria. The identifiers can be, for example, identifiers of nodes of the conversation tree representing those emails.

[0129] This set of emails of the relative scope can then be iterated through to determine if any other of the set of emails in the relative scope subsume that email. For a node of the conversation tree representing an email of the relative scope (the email and node under evaluation), a subsumed by set for the email represented by that node may be determined (STEP **406**). For example, this subsumed by set may be the set of identifiers of other emails of the conversation that subsumes the email represented by the node under evaluation. In particular, this subsumed by set may be stored in association with the node representing the node under evaluation and include identifiers of other nodes of the conversation tree that represent emails that subsume the email associated with the

node under evaluation.

[0130] The intersection of the relative scope set can then be intersected with the subsumed by set (STEP **408**). In particular, the set of subsumed by identifiers for the node being evaluated can be intersected with the relative scope set (comprising identifiers of emails of the relative scope of the conversation). If the intersection of the subsumed by set for the node being evaluated and the relative scope set is empty (Y Branch of STEP **410** indicating that the email being evaluated is not subsumed by any other email included in the relative scope) the (e.g. identifier of the) email being evaluated may be added to the final set of emails for that conversation (STEP **412**). If the intersection of the subsumed by set for the node being evaluated and the relative scope set is not empty (N Branch of STEP **410** indicating that the email being evaluated is subsumed by at least one other email included in the relative scope) set the identifier of the email may not be added to the final set of emails for that conversation.

[0131] If there are nodes of the conversation tree that have not been evaluated (Y Branch of STEP **414**), the next node in conversation tree may be obtained and evaluated. If there are no remaining nodes of the conversation tree to evaluate (N Branch of STEP **414**), the final set of emails responsive to all scoping criteria (e.g., the one or more criteria and the subsumed or inclusive criteria) can then be returned (STEP **416**) (e.g., for presentation to a user in an interface for that conversation tree).

[0132] It may now be useful to an understanding of embodiments to illustrate a brief example. For purpose of this illustration reference is made back briefly to the email conversation **210** of FIG. **2**C comprising the six emails **220** along with FIG. **3**C depicting conversation tree **380***a* representing email conversation **210**. Recall that in the example of FIG. **2**C the user specified scoping criteria **224** comprising a text term "pumpkin" and a date range of Oct. 1, 2021 to Oct. 31, 2021. Assume additionally that the user also specifies as scoping criteria that subsumed emails are not be included in emails returned for conversations.

[0133] Looking now at conversation tree **380***a* of FIG. **3**C, to determine the final set of emails to return to a user for conversation tree **380***a* representing email conversation **210**, in one embodiment, a set of (identifiers for) nodes **374** of conversation tree **380***a* for a relative scope may be determined. Here, the relative scope of the email conversation **210** should comprise any emails (e.g., including attachments) that include the term "pumpkin" and were sent (e.g., or existing, received, etc.) on or between Oct. 1, 2021 to Oct. 31, 2021.

[0134] Continuing with the example elaborated on above, assume that attachment **222***a*, the text content of the third email **220***c* and the text content of the fourth email **220***d* all include the term "pumpkin". Thus, first email **220***a*, second email **220***b*, third email **220***c* and fourth email **220***d* may be deemed part of the relative scope **226** of the email conversation determined based on the scoping criteria as these emails **220** include both the term "pumpkin" and fall within the data range of the scoping criteria **224**. Conversely, while fifth and sixth email **220***e*, **220***f* may include the term "pumpkin", they fall outside the data range specified in the scoping criteria and are thus not in the relative scope of email conversation **210** as determined using the scoping criteria.

[0135] Thus, using conversation tree **380***a* the relative scope set for the scoping criteria in question may be determined to comprise (identifiers for) nodes **374***d*, **374***c*, **374***b* and **374***a*. The relative scope set may thus be: [1, 2, 3, 4]. Each of the nodes of that relative scope set may then be iterated through to see if the email corresponding to that node **374** is subsumed by another email associated with conversation tree **380***a*.

[0136] Starting with node **1 374***a* representing the first email **220***a* the subsumed by set for this email may be determined as [2] based on the subsumed by field associated with that node **374***a*. Intersecting the relative scope set [1, 2, 3, 4] with the subsumed by set for node **1 374***a* [2] yields a non-empty set (e.g., [2]). Thus, email **220***a* represented by node **1 374***a* is not added to the final set of emails for conversation tree **380***a*.

[0137] Moving to node **2 374***b* representing the second email **220***b* of conversation **210**, the

subsumed by set for this email may be determined as [ ](empty) based on the subsumed by field associated with that node **374***b*. Intersecting the relative scope set [1, 2, 3, 4] with the subsumed by set for node **2 374***b* [ ] yields an empty set [ ]. As such email **220***b* represented by node **1 374***b* will be added to the final set of emails for conversation tree **380***a*.

[0138] With respect to node **3 374***c* representing the third email **220***c* of conversation **210**, the subsumed by set for this email may determined as [4, 5, 6] based on the subsumed by field associated with that node **374***c*. Intersecting the relative scope set [1, 2, 3, 4] with the subsumed by set for node **3 374***c* [4, 5, 6] yields a non-empty set (e.g., [4]). Here, email **220***c* represented by node **1 374***c* is not added to the final set of emails for conversation tree **380***a*.

[0139] Lastly, for node **4 374***d* representing the fourth email **220***b* of conversation **210** the subsumed by set for this email may be determined as [5, 6] based on the subsumed by field associated with that node **374***d*. Intersecting the relative scope set [1, 2, 3, 4] with the subsumed by set for node **4 374***d* [5, 6] yields an empty set [ ]. Consequently, email **220***d* represented by node **4 374***d* will be added to the final set of emails for conversation tree **380***a*.

[0140] As can be seen then, by utilizing embodiment as discussed herein, the final set of emails to return for conservation tree **380***a* can be determined to include second email **220***b* (including attachments **222***a*, **222***b*) and fourth email **220***d*, as these two emails **220***b*, **220***d* encompass are the minimum set of emails that encompass all the content (e.g., text and attachments **222**) included in the relative scope determined based on the scoping criteria.

[0141] Jumping back now to FIG. **1**, as the email graph **172** of document analysis system may be involved with many operations it may be desired to efficiently operate on (e.g., create, update, alter, modify or otherwise access or interact with) email graph **172**. To illustrate in more detail, as discussed, in many scenarios email data may be retrieved or obtained from different source systems (e.g., email servers, which may be of different types) at different times. Thus, the conversation trees of the email graph **172** (e.g., simplified graph **115**) determined at a particular point may be incomplete. For example, all the email data about such conversations may not yet have been obtained from source systems having such data, and new email data regarding such conversation trees may be obtained at (e.g., multiple) subsequent times. Accordingly, such document analysis system **101** may be required to frequently update the simplified graph **115**. Specifically, it may be desired to adapt embodiments to identify conversation trees pertaining to newly obtained email data and update the identified conversation trees in the simplified graph **115** (and, additionally, the raw graph **113**).

[0142] However, because the simplified graph **115** may collapse or eliminate some of the data of the raw graph **113** (e.g., weak reply relationships, email nodes, etc.) a problem may now occur in which the simplified graph does not contain enough data to facilitate the efficient incremental update of the simplified graph **115** when such new email data is obtained. To understand such issues, along with the embodiments of incrementally updating such raw graphs and simplified graphs, one embodiment of a table structure that may be utilized in a columnar data store is depicted in FIG. **3**D. Embodiments of such a table structure may be a functionally recursive manner of encoding a graph structure that matches well with the capabilities of a columnar data store, including, for example, common table expressions or recursive common table expressions. In this example, the matter identifier (e.g., the "matter_uuid") may correspond to an identifier associated with a tenant of the document analysis system, while the "conv_id" may correspond to an identifier for a conversation within a graph (e.g., an identified conversation tree). An instance (e.g., column or row) of a "final_graph" table may correspond to an email node while an instance (e.g., column or row) of a "edges" may correspond to a reply relationship.

[0143] Moving back to FIG. **1**, accordingly, in certain embodiments, when new email data is received at the document analysis system **101** and incremental update of the email graph of the document analysis system **101** may be performed, whereby an incremental graph may be determined and this incremental graph utilized to update the email graph. In some embodiments,

conversation service **124** may utilize the raw graph **113** to perform an incremental update to the simplified graph **115**. These incremental updates may service to apply pinpoint edits to update only the portions (e.g., conversations) of the raw graph **113** and simplified graph **115** that need to be updated based on the new data without having to rebuild, reevaluate or traverse the entirety of the raw graph **113** or the simplified graph **125**.

[0144] In certain embodiments then, the conversation service **124** may determine the incremental updates to the raw graph **113** and the simplified graph **115** based on the new mail data and utilize the database system **105** (including the database engine **109**) to preform incremental updated to the raw graph **113** and the simplified graph **115** in an efficient manner that may be data local (e.g., performed as close to the data in the database system **105** as possible, such as in database engine **109**), distributed (e.g., using the parallel processing capabilities of database system **105** for instance) and cross-tenant (e.g., in a manner that can be utilized to preform incremental updates to data across tenant data in the raw graph **113** and simplified graph **115** when possible. To illustrate, as the raw graph **113** and simplified graph **115** may represent the email data **107** of multiple tenants of the document analysis system the overhead to process or update raw graph **113** or simplified graph **115** for multiple tenants may be shared when processing data for those graphs across all tenants.

[0145] According to embodiments, then, the newly obtained email data may be loaded or otherwise accessed. New nodes or edges to add can then be determined as part of building an incremental graph comprising an incremental raw graph **123** and a corresponding incremental simplified graph **125**. The emails of the incremental simplified graph **125** can then be evaluated to determine which of the emails of the simplified graph are inclusive (or subsumed) and those emails (e.g., the nodes representing those emails) may be annotated or flagged (e.g., with identifiers of emails or nodes which are included or inclusive accordingly, etc.). This incremental simplified graph **125** can then be used by conversation service **124** (e.g., in the presentation of an interface displaying an email conversation or otherwise). Additionally, the incremental raw graph **123** can be used to update or replace the corresponding portions (e.g., conversations) in the raw graph **123** and the incremental simplified graph **125** can be used to update or replace the corresponding portions (e.g., conversations) in the simplified graph **115**.

[0146] Thus, first an incremental raw graph **123** may be determined from newly received email data. Such an incremental raw graph **123** may be determined utilizing instructions sent from the conversation service **124** to the database engine **109** and may be maintained or stored in tables at the database system **105** (e.g., for example, in memory of the database engine **109**). Specifically, one or more nodes may be determined to add to the incremental raw graph **123** based on the identification of a new email from email data **107** (e.g., either obtained new email metadata or an obtained new email document). Alternatively, a node to add may represent newly identified email data (e.g., an email document or email metadata) pertaining to an email already represented by one or more nodes of the existing raw graph **113**. These new nodes to add to the incremental raw graph may then be added to the incremental raw graph **113**.

[0147] New edges to add to the raw graph **113** may be determined and added to the incremental raw graph **123** based on the email data associated with (e.g., each of) the newly determined node. These new edges may include new equivalency edges representing other existing nodes in the raw graph **113** that represent email data for the same email or a reply relationship edge between the new node and a second node of the raw graph **113** representing another email. Specifically, the newly determined equivalency edges or reply relationships may be determined based on email metadata received as part of the new email data **107**. Such equivalencies between emails and reply relationships between emails may be specified in such email metadata and may be determined and added to the incremental raw graph **113** along with any nodes of the raw graph **113** related to the new node by the added relationship.

[0148] Edges to add to the incremental raw graph **123** may also be determined from the content of

the email documents themselves. To determine such content edges, for the newly added email node of the incremental raw graph **113**, all content edge candidates can be determined. These content edge candidates may be determined by paring the email represented by the newly added node to all emails (e.g., in the email data **107** associated with the same tenant, etc.) that share a (e.g., similar) subject line and compatible send dates in the new email data **107**. Existing (e.g., previously received) emails may also be included (e.g., if they share a subject with the email represented by the newly added node). Here, for example, if a normalized subject of one email matches up to another email and an extracted send date for a text segment of one of the emails for the body of that email matches a second send date for a text segment **121** of from a quoted portion of another email a candidate content edge between the two emails may be created in the candidate content edge table.

[0149] These content edge candidates (e.g., pairs of email which are candidates to having their representative nodes linked by a reply relationship) can thus be stored in a table (e.g., at the database engine **109**). The content edge candidate table may then be joined with the table of text segments **121** extracted for each of the emails of each of the emails referenced in the content edge candidate table. By joining the tables in this manner if comparison of the text segments is needed this comparison may be done in an efficient manner.

[0150] A similarity metric for each of the content edge candidates can then be determined. Specifically, the text segments of each pair of emails for the candidate content edge can then be compared. In one embodiment for each pair, a text segment **121** for the body (or a deeper level of quotation) of one email associated with candidate content edge may be compared to a second text segment **121** for a first (or deeper level) level of quotation of the other email associated with candidate content edge. This comparison may be done in both directions for the emails of the candidate content edge. In other words, the similarity metric can assess the similarity between text of one email for the candidate content edge and quoted text in the other email to determine if one of the emails associated with the candidate content edge has been quoted in the other email of the content edge. This comparison can be done using a Levenshtein distance metric (e.g., the levenshtein_ratio function). This similarity metric can then be stored for that candidate content edge in the content edge table. This similarity metric can be used to select a best candidate content edge for a new node based on the similarity metric (e.g., based on the highest determined similarity). Such a selection can be determined, for example, by a windowing function or the like. The selected candidate content edge (and the corresponding similarity metric) can be inserted into the incremental raw graph **123** (with the similarity metric value being assigned as the weight to the inserted content edge). The incremental raw graph **123** thus includes the new node and any determined equivalence edges or relationship edge along with the other nodes related to the new node by the determined edges.

[0151] The remainder of the incremental raw graph **123** may then be determined and the simplified graph **125** determined from the incremental raw graph **123**. Specifically, each of the edges of the incremental raw graph **123** can then be followed recursively using the raw graph **113**, and each node and relationship transitively connected to an of the edge in the incremental raw graph **123** in the raw graph **113** may be added to the incremental raw graph **123**. This recursive addition may serve to add adjacent edges and equivalent edges from the raw graph **113** recursively. Here, each edge in the incremental raw graph **123** can be accessed in the raw graph **113** and followed to the next node, this node can then be added to the incremental raw graph **123**, and each edge of that new node recursively added to the incremental raw graph **123**. Thus, if an accessed node in the raw graph **113** is a root node or a leaf node it will be added to the incremental raw graph **123** and no further edges may exist in the raw graph **113** to follow. Thus, at this point the incremental raw graph **123** may include each newly determined node from the newly received email data **107**. Moreover, for each newly determined node, the incremental raw graph **123** may include all nodes that of the raw graph **113** that are related (directly or indirectly) to the new node (e.g., based on the

initial determined edges for that new node as determined through, for example, a textual comparison of emails or through email metadata).

[0152] Once the incremental raw graph **123** is determined it can be transformed into incremental simplified graph **125** by collapsing the equivalent nodes of the incremental raw graph **123** to get a single node in incremental simplified graph **125** representing all equivalent nodes in the incremental raw graph **123**. This can be done, for example, using a COALESCE function or the like. Additionally, a best reply relationship edge may be selected for the simplified graph **125** between any two nodes based on the sum of the weights of all the reply relationship edges between all equivalent nodes for those two nodes (e.g., the nodes collapsed to form those two nodes) in the incremental raw graph **123**. The weights associated with the reply edges of the incremental raw graph **123** can be used to select the best reply edges based on the similarity metric (e.g., based on the highest determined similarity) associated with each edge. Such a selection can be determined, for example, by a windowing function or the like. The determined best reply relationship edges are then added to the incremental simplified graph **125**.

[0153] This incremental simplified graph **125** can then be traversed to determine conversations within the incremental simplified graph and assign identifiers to all the nodes of each conversation tree within the incremental simplified graph **125**. Specifically, the incremental simplified graph **125** can be obtained and recursively traversed from the root node of each conversation tree in the simplified graph **125** to the leaves of each conversation tree. Thus, such a traversal may determine each node of the incremental simplified graph that is a root node (e.g., a node that has no reply relationship edges specifying that node as a reply and representing an originator email) for a conversation tree and beginning with each of those nodes traverse the associated conversation tree originating with the node. This traversal may serve to assign a conversation identifier to the conversation represented by the conversation tree associated with each root node and, in some embodiments, may impose a size limit to the conversations. Such a traversal may, in one embodiment, be done by a recursive common table expression (CTE) from the root nodes down to the leaves of a conversation tree in order to efficiently implement such a traversal.

[0154] Starting with the root node of the conversation tree, each child node of the root node may be determined. Each of the child nodes can then be updated with the conversation identifier in the incremental simplified graph **125**. In one embodiment, a conversation identifier may be, for example, an identifier of the root node (e.g., representing an originator email) of the conversation tree, such that the identifier for the root node is propagated to each of the nodes of that conversation tree subgraph during traversal.

[0155] Additionally, in one embodiment a node level of the traversal may be tracked and incremented. Such a node level may be added to the node (e.g., as an attribute or property of the node) in the incremental simplified graph **125**. For each of the child nodes it can be determined if that child node is a leaf node. If the node is a leaf node, it can be determined if there are any more nodes in the node list. If there are no more nodes in the node list the recursive traversal may terminate, while if there are more nodes in the node list, the remining nodes in the node list may be recursively traversed. If the node is not a leaf node all the child nodes of that node may be added to a list of nodes, and any remining nodes in the node list recursively traversed. In one embodiment, the number of nodes in the current conversation tree may be tracked (e.g., by grouping the rows of a table comprising the incremental simplified graph **125**) to determine if the number of nodes in the conversation tree exceeds some conversation size limit (e.g., 1000 nodes) and if the conversation size limit has been exceeded the traversal may stop.

[0156] Once the traversal of the conversation trees of the incremental simplified graph is terminated, the updated incremental simplified graph **125** may be determined and output. This incremental traversed simplified graph **125** can then be evaluated to detect inclusive emails. Emails that have content (e.g., text alone, text and attachments, etc.) that is fully duplicated by other emails later in their thread are called subsumed. All other emails may be called inclusive. Knowledge of

which emails are inclusive or subsumed by other emails may be used by the document analysis system **101** as previously discussed, such as by determining emails of a conversation to present and to help users of the document analysis system **101** work efficiently by ignoring (or not being presented with) subsumed emails and only looking at those that are inclusive.

[0157] Thus, the traversed incremental simplified graph **125** can then be used to detect inclusive or subsumed emails in embodiments. Information indicative of which other emails (if any) subsume an email can then be indicated in nodes if the simplified graph **125**. According to embodiments then, once the traversed incremental simplified graph **125** is obtained, it can be traversed from the leaf nodes to the root node of the conversation tree. Specifically, for a conversation tree in the incremental simplified graph **125** each of the branches of the conversation tree may be identified (e.g., where a node of the conversation tree may be associated with one or more branches). Starting at a leaf node of the conversation tree, that branch may be followed upward until the root node of the conversation tree is reached, identifying each node of that branch with a branch identifier (e.g., which may be a node identifier of the leaf node which is the termination of that branch). Again, such a traversal may be accomplished with a recursive common table expression. Accordingly, each of the branches (e.g., comprising a set of nodes and relationships of a path through the conversation tree between a leaf node of the conversation tree and the root node of the conversation tree) of the conversation tree may be identified and placed in a branch table.

[0158] The text of the emails of each identified branch of the conversation tree can then be analyzed to determine if the text of any email represented by a node of the branch is duplicated in any descendant of the branch. Based on this determination, the nodes of the branch may be annotated indicated if the email represented by the node is inclusive or subsumed. Specifically, in one embodiment, a node may be annotated (e.g., the subsumed by field of the node) with identifiers of nodes representing emails that subsume that email. To accomplish this, the branch table with the branches for the conversation tree may be joined with the text segments **121** to facilitate comparison. Each node of a branch can then be paired with all of its descendants. For example, a self-join leveraging the branch identifier for the branch and a node level for each node (e.g., as identified during a previous traversal as discussed) each node may be paired with all of its descendants. The emails associated with each of the paired nodes (e.g., a node and its descendants) can then be checked by comparing the text segments associated with each of the emails associated with those nodes to determine if the text of the email represented by one of the nodes is duplicated in any of the descendant nodes. If the text is duplicated the node representing the email which includes the duplicated content may be indicated (e.g., flagged or otherwise annotated) as inclusive. Embodiments may maintain a single flag indicating whether that email is subsumed or not. Additionally subsumed by (or inclusive) information may be recorded for nodes representing inclusive (or subsumed) emails (or emails which subsume other emails) in the conversation tree. In particular, according to certain embodiments, descendant or parent information can be stored in association with a node representing a subsumed email that indicate which other emails in that conversation subsume the content of that email, or conversely for nodes representing an email which subsumes another email which other email(s) are subsumed by that email.

[0159] This annotated traversed incremental simplified graph can then be output for use in presenting an interface corresponding to any of the conversations trees represented in the incremental simplified graph **125**. Moreover, the incremental raw graph **123** and the incremental simplified graph **125** may then be used to update the raw graph **113** and simplified graph **115** respectively such that both the raw graph **113** and the simplified graph **115** may be incrementally updated based on newly received email data. Specifically, the subgraphs of the incremental raw graph **113** and the annotated conversation trees of the incremental simplified graph **125** may be substituted for the corresponding subgraphs and conversation trees in the raw graph **113** and the simplified graph **115**, respectively.

[0160] In this manner, then, the document analysis system (and thus users of such a document

analysis system) may quickly obtain, analyze, or otherwise utilize, new email data, without the need to reconstruct or traverse the entirety of data structures (e.g., graphs) representing the entirety of the email data obtained from source systems. Instead, such data structures may be efficiently and incrementally updated utilizing incremental data structures representing only portions of the data structures that will need to be altered based on the newly obtained data.

[0161] Moreover, through the use of the encoding of the data structures utilized by embodiments and the data stores or engines used to store and manipulate such data structures, embodiments may provide the advantages that the processing of email data and the manipulation of the representations of email conversations may be efficiently implemented in a distributed or data-local manner using a database engine that may be part of a database platform such as a database as a service. Additionally, in instances where the document analysis system is a multi-tenant platform, the processing may be accomplished as single (or fewer) job executions that can be more efficiently processed for multiple tenants.

[0162] Moving on to FIG. **5**, one embodiment of a method for the incremental update of email graphs that may be employed by a document analysis system is depicted. Newly obtained email data may be loaded or otherwise obtained (e.g., accessed or received) (STEP **502**). New nodes or edges to add can then be determined (STEP **504**) as part of building an incremental raw graph and a corresponding incremental simplified graph (STEP **506**). The emails of the incremental simplified graph can then be evaluated to determine which of the emails of the simplified graph are inclusive (or subsumed) and those emails (e.g., the nodes representing those emails) may be flagged or otherwise annotated (e.g., with identifiers of emails or nodes which are included or inclusive accordingly, etc.) (STEP **508**). This incremental simplified graph can then be used in the presentation of an interface displaying an email conversation or otherwise (STEP **510**). Additionally, the incremental raw graph can be used to update or replace the corresponding portions (e.g., conversations) in the raw graph (STEP **512**) and the incremental simplified graph can be used to update or replace the corresponding portions (e.g., conversations) in the simplified graph (STEP **514**).

[0163] Initially, then, once new email data is obtained from one or more source systems an incremental raw graph may be determined. Turning to FIG. **6**, one embodiment of a method for deterring content edges to add to the incremental raw graph is depicted. As a first step, one or more nodes may be determined to add to the incremental raw graph (STEP **602**). This new node to add may be based on the identification of a new email from email data (e.g., either obtained new email metadata or an obtained new email document). Alternatively, a new node to add may represent newly identified email data (e.g., an email document or email metadata) pertaining to an email already represented by one or more nodes of the existing raw graph. These new nodes to add to the incremental raw graph may then be added to the incremental raw graph.

[0164] New edges to add to the raw graph may be determined and added to the incremental raw graph based on the email data associated with (e.g., each of) the newly determined node. These new edges may include new equivalency edges representing other existing nodes in the raw graph that represent email data for the same email or a reply relationship edge between the new node and a second node of the raw graph representing another email. The newly determined equivalency edges or reply relationships may be determined based on email metadata received as part of the new email data (STEP **604**). Such equivalencies between emails and reply relationships between emails may be specified in such email metadata and may be determined and added to the incremental raw graph along with any nodes of the raw graph related to the new node by the added relationship.

[0165] Edges to add to the incremental raw graph may also be determined from the content of the email documents themselves (STEP **612**). To determine such content edges, for the newly added email node of the incremental raw graph, all content edge candidates can be determined (STEP **606**). These content edge candidates may be determined by paring the email represented by the

newly added node to all emails (e.g., in the email data associated with the same tenant, etc.) that share a (e.g., similar) subject line and compatible send dates in the new email data. Existing (e.g., previously received) emails may also be included (e.g., if they share a subject with the email represented by the newly added node). Here, for example, if a normalized subject of one email matches up to another email and an extracted send date for a text segment of one of the emails for the body of that email matches a second send date for a text segment of from a quoted portion of another email a candidate content edge between the two emails may be created in the candidate content edge table.

[0166] These content edge candidates (e.g., pairs of email which are candidates to having their representative nodes linked by a reply relationship) can thus be stored in a table. The content edge candidate table may then be joined with the table of text segments extracted for each of the emails of each of the emails referenced in the content edge candidate table (STEP **608**). By joining the tables in this manner if comparison of the text segments is needed this comparison may be done in an efficient manner.

[0167] A similarity metric for each of the content edge candidates can then be determined (STEP **610**). Specifically, the text segments of each pair of emails for the candidate content edge can then be compared. In one embodiment for each pair, a text segment for the body (or a deeper level of quotation) of one email associated with candidate content edge may be compared to a second text segment for a first (or deeper level) level of quotation of the other email associated with candidate content edge. This comparison may be done in both directions for the emails of the candidate content edge. In other words, the similarity metric can assess the similarity between text of one email for the candidate content edge and quoted text in the other email to determine if one of the emails associated with the candidate content edge has been quoted in the other email of the content edge. This comparison can be done, for example, using a Levenshtein distance metric (e.g., the levenshtein_ratio function). This similarity metric can then be stored for that candidate content edge in the content edge table.

[0168] This similarity metric can be used to select candidate content edges (e.g., including a best candidate content edge) for a new node based on the similarity metric (e.g., based on the highest determined similarity) (STEP **614**). Such a selection can be determined, for example, by a windowing function or the like based on, for example, if the similarity metric is over some threshold value (e.g., 90% or the like). The selected candidate content edge (and the corresponding similarity metric) can be inserted into the incremental raw graph (with the similarity metric value being assigned as the weight to the inserted content edge) (STEP **616**). The incremental raw graph thus includes the new node and any determined equivalence edges or relationship edge along with the other nodes related to the new node by the determined edges.

[0169] It may be useful here to briefly illustrate visually such a comparison of text segments. FIG. **7** depicts a first email **702***a* and a second email **702***b*, along with text segments **710** (e.g., including for example, headers, normalized subjects etc.) that may be extracted from these emails. Notice that text segment **1 710***a* may be the body of the first email, while text segment **2 710***b* may be a first level quoted portion of second email **702***b*. Thus, by comparing the subject **710***e* of first email **702***a* with the subject **710***f* of the second email **702***b*, the send date **710***c* of first email **702***a* with the extracted send date **710***d* of the second email **702***b* associated with the first level quoted portions of the second email **702***b*, and text segment **1 710***a* with text segment **710***b* of a first level quoted portion of second email **702***b* as detailed in the enumerated steps 1-4 of the embodiment depicted it can be determined if a content edge should be added to an incremental raw graph.

[0170] Moving to FIG. **8**, one embodiment of a method for building the remainder of the incremental graphs is depicted. Here, once the edges are determined and added to the incremental raw graph (e.g., as depicted in FIG. **7**) the remainder of the incremental raw graph may then be determined and the simplified graph determined from the incremental raw graph. Specifically, each of the edges of the incremental raw graph can then be followed recursively using the raw graph,

and each node and relationship transitively connected to an of the edge in the incremental raw graph in the raw graph may be added to the incremental raw graph (STEPS **802**, **804**). This recursive addition may serve to add adjacent edges and equivalent edges from the raw graph recursively. Here, each edge in the incremental raw graph can be accessed in the raw graph and followed to the next node, this node can then be added to the incremental raw graph, and each edge of that new node recursively added to the incremental raw graph. Thus, if an accessed node in the raw graph is a root node or a leaf node it will be added to the incremental raw graph and no further edges may exist in the raw graph to follow. At this point the incremental raw graph may include each newly determined node from the newly received email data. Moreover, for each newly determined node, the incremental raw graph may include all nodes that of the raw graph that are related (directly or indirectly) to the new node (e.g., based on the initial determined edges for that new node as determined through, for example, a textual comparison of emails or through email metadata).

[0171] Once the incremental raw graph is determined it can be used to determine incremental simplified graph (STEP **806**) by collapsing the equivalent nodes of the incremental raw graph to get a single node in incremental simplified graph **125** representing all equivalent nodes in the incremental raw graph (STEP **808**). This can be done, for example, using a COALESCE function or the like. Additionally, a best reply relationship edge may be selected for the simplified graph between any two nodes based on the sum of the weights of all the reply relationship edges between all equivalent nodes for those two nodes (e.g., the nodes collapsed to form those two nodes) in the incremental raw graph (STEP **810**). The weights associated with the reply edges of the incremental raw graph can be used to select the best reply edges based on the similarity metric (e.g., based on the highest determined similarity) associated with each edge. Such a selection can be determined, for example, by a windowing function or the like. The determined best reply relationship edges are then added to the incremental simplified graph.

[0172] This incremental simplified graph can then be traversed to determine conversations within the incremental simplified graph and assign identifiers to all the nodes of each conversation tree within the incremental simplified graph. One embodiment for a method for traversing the incremental graph is depicted in FIG. **9**. In this embodiment, the incremental simplified graph can be obtained (STEP **902**) and recursively traversed from the root node of each conversation tree in the simplified graph to the leaves of each conversation tree (STEP **920**). Thus, such a traversal may determine each node of the incremental simplified graph that is a root node (e.g., a node that has no reply relationship edges specifying that node as a reply and representing an originator email) for a conversation tree and beginning with each of those root nodes traverse the associated conversation tree originating with the node. This traversal may serve to assign a conversation identifier to the conversation represented by the conversation tree associated with each root node and, in some embodiments, may impose a size limit to the conversations. Such a traversal may, in one embodiment, be done by a recursive common table expression (CTE) from the root nodes down to the leaves of a conversation tree in order to efficiently implement such a traversal.

[0173] Starting with the root node of the conversation tree then (STEP **904**), each child node of the root node may be determined and added to a node list. For each of the nodes in the node list, that node can then be updated with the conversation identifier in the incremental simplified graph (STEP **906**). In one embodiment, a conversation identifier may be, for example, an identifier of the root node (e.g., representing an originator email) of the conversation tree, such that the identifier for the root node is propagated to each of the nodes of that conversation tree subgraph during traversal. Additionally, in one embodiment a node level of the traversal may be tracked and incremented. Such a node level may be added to the node (e.g., as an attribute or property of the node) in the incremental simplified graph.

[0174] It can be determined if that node is a leaf node (STEP **910**). If the node is a leaf node (Y branch of STEP **910**), it can be determined if there are any more nodes in the node list (STEP **914**).

If there are no more nodes in the node list (N branch of step **914**) the recursive traversal may terminate, while if there are more nodes in the node list (Y branch of STEP **914**), the remaining nodes in the node list may be recursively traversed. If the node is not a leaf node (N branch of STEP **910**) all the child nodes of that node currently being processed may be added to a list of nodes (STEP **912**), and any remaining nodes in the node list recursively traversed (STEP **905**). In one embodiment, the number of nodes in the current conversation tree may be tracked (e.g., by grouping the rows of a table comprising the incremental simplified graph) to determine if the number of nodes in the conversation tree exceeds some conversation size limit (e.g., 1000 nodes) and if the conversation size limit has been exceeded the traversal may stop. Once the traversal of the conversation trees of the incremental simplified graph is terminated, the updated incremental simplified graph may be determined and output (STEP **922**).

[0175] This incremental traversed simplified graph can then be evaluated to detect inclusive emails. FIG. **10** depicts one embodiment of a method for detecting inclusive emails using the incremental simplified graph. Once the traversed incremental simplified graph is obtained (STEP **1002**), it can be traversed from the leaf nodes to the root node of the conversation tree (STEP **1020**). Specifically, for a conversation tree in the incremental simplified graph each of the branches of the conversation tree may be identified (e.g., where a node of the conversation tree may be associated with one or more branches) (STEP **1004**). Starting at a leaf node of the conversation tree, that branch may be followed upward until the root node of the conversation tree is reached, identifying each node of that branch with a branch identifier (e.g., which may be a node identifier of the leaf node which is the termination of that branch). Again, such a traversal may be accomplished with a recursive common table expression. Accordingly, each of the branches (e.g., comprising a set of nodes and relationships of a path through the conversation tree between a leaf node of the conversation tree and the root node of the conversation tree) of the conversation tree may be identified and placed in a branch table.

[0176] The text of the emails of each identified branch of the conversation tree can then be analyzed to determine if the text of any email represented by a node of the branch is duplicated in any descendant of the branch. Based on this determination, the nodes of the branch may be annotated indicated if the email represented by the node is inclusive or subsumed. To accomplish this, the branch table with the branches for the conversation tree may be joined with the text segments to facilitate comparison. Each node of a branch can then be paired with all of its descendants (STEP **1006**). For example, a self-join leveraging the branch identifier for the branch and a node level for each node (e.g., as identified during a previous traversal as discussed) each node may be paired with all of its descendants. The emails associated with each of the paired nodes (e.g., a node and its descendants) can then be checked to determine if an email is duplicated in any descendant email (STEP **1008**). This determination can be made by comparing the text segments associated with each of the emails associated with those nodes to determine if the text of the email represented by one of the nodes is duplicated in any of the descendant nodes (STEP **1010**). If the text is duplicated (Y branch of STEP **1010**) the node representing the email which includes the duplicated content may be indicated (e.g., flagged or otherwise annotated) as inclusive (STEP **1012**). Otherwise, other nodes of the branches may be processed.

[0177] In one embodiment, indicating a node as inclusive or subsumed may include recording information for nodes representing inclusive (or subsumed) emails (or emails which subsume other emails) in the conversation tree. In particular, according to certain embodiments, descendant or parent information can be stored in association with a node representing a subsumed email that indicate which other emails in that conversation subsume the content of that email, or conversely for nodes representing an email which subsumes another email which other email(s) are subsumed by that email.

[0178] This annotated traversed incremental simplified graph can then be output for use in presenting an interface corresponding to any of the conversations trees represented in the

incremental simplified graph (STEP **1022**). Moreover, the incremental raw graph and the incremental simplified graph may then be used to update the raw graph and simplified graph respectively such that both the raw graph and the simplified graph may be incrementally updated based on newly received email data. Specifically, the subgraphs of the incremental raw graph and the annotated conversation trees of the incremental simplified graph may be substituted for the corresponding subgraphs and conversation trees in the raw graph and the simplified graph, respectively.

[0179] Although the invention has been described with respect to specific embodiments thereof, these embodiments are merely illustrative, and not restrictive of the invention. The description herein of illustrated embodiments of the invention, including the description in the Abstract and Summary, is not intended to be exhaustive or to limit the invention to the precise forms disclosed herein. Rather, the description is intended to describe illustrative embodiments, features and functions in order to provide a person of ordinary skill in the art context to understand the invention without limiting the invention to any particularly described embodiment, feature, or function, including any such embodiment feature or function described in the Abstract or Summary. While specific embodiments of, and examples for, the invention are described herein for illustrative purposes only, various equivalent modifications are possible within the spirit and scope of the invention, as those skilled in the relevant art will recognize and appreciate. As indicated, these modifications may be made to the invention in light of the foregoing description of illustrated embodiments of the invention and are to be included within the spirit and scope of the invention. Thus, while the invention has been described herein with reference to particular embodiments thereof, a latitude of modification, various changes and substitutions are intended in the foregoing disclosures, and it will be appreciated that in some instances some features of embodiments of the invention will be employed without a corresponding use of other features without departing from the scope and spirit of the invention as set forth. Therefore, many modifications may be made to adapt a particular situation or material to the essential scope and spirit of the invention.

[0180] Reference throughout this specification to "one embodiment", "an embodiment", or "a specific embodiment" or similar terminology means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment and may not necessarily be present in all embodiments. Thus, respective appearances of the phrases "in one embodiment", "in an embodiment", or "in a specific embodiment" or similar terminology in various places throughout this specification are not necessarily referring to the same embodiment. Furthermore, the particular features, structures, or characteristics of any particular embodiment may be combined in any suitable manner with one or more other embodiments. It is to be understood that other variations and modifications of the embodiments described and illustrated herein are possible in light of the teachings herein and are to be considered as part of the spirit and scope of the invention.

[0181] In the description herein, numerous specific details are provided, such as examples of components or methods, to provide a thorough understanding of embodiments of the invention. One skilled in the relevant art will recognize, however, that an embodiment may be able to be practiced without one or more of the specific details, or with other apparatus, systems, assemblies, methods, components, materials, parts, and/or the like. In other instances, well-known structures, components, systems, materials, or operations are not specifically shown or described in detail to avoid obscuring aspects of embodiments of the invention. While the invention may be illustrated by using a particular embodiment, this is not and does not limit the invention to any particular embodiment and a person of ordinary skill in the art will recognize that additional embodiments are readily understandable and are a part of this invention.

[0182] Embodiments discussed herein can be implemented in a computer communicatively coupled to a network (for example, the Internet), another computer, or in a standalone computer. As is known to those skilled in the art, a suitable computer can include a CPU, at least one read-only

memory ("ROM"), at least one random access memory ("RAM"), at least one hard drive ("HD"), and one or more input/output ("1/O") device(s). The I/O devices can include a keyboard, monitor, printer, electronic pointing device (for example, mouse, trackball, stylus, touch pad, etc.), or the like.

[0183] ROM, RAM, and HD are computer memories for storing computer-executable instructions executable by the CPU or capable of being compiled or interpreted to be executable by the CPU. Suitable computer-executable instructions may reside on a computer readable medium (e.g., ROM, RAM, and/or HD), hardware circuitry or the like, or any combination thereof. Within this disclosure, the term "computer readable medium" is not limited to ROM, RAM, and HD and can include any type of data storage medium that can be read by a processor. For example, a computer-readable medium may refer to a data cartridge, a data backup magnetic tape, a floppy diskette, a flash memory drive, an optical data storage drive, a CD-ROM, ROM, RAM, HD, or the like. The processes described herein may be implemented in suitable computer-executable instructions that may reside on a computer readable medium (for example, a disk, CD-ROM, a memory, etc.). Alternatively, the computer-executable instructions may be stored as software code components on a direct access storage device array, magnetic tape, floppy diskette, optical storage device, or other appropriate computer-readable medium or storage device.

[0184] Any suitable programming language can be used to implement the routines, methods, or programs of embodiments of the invention described herein, including C, C++, J ava, JavaScript, HTML, or any other programming or scripting code, etc. Other software/hardware/network architectures may be used. For example, the functions of the disclosed embodiments may be implemented on one computer or shared/distributed among two or more computers in or across a network. Communications between computers implementing embodiments can be accomplished using any electronic, optical, radio frequency signals, or other suitable methods and tools of communication in compliance with known network protocols.

[0185] Different programming techniques can be employed such as procedural or object oriented. Any particular routine can execute on a single computer processing device or multiple computer processing devices, a single computer processor or multiple computer processors. Data may be stored in a single storage medium or distributed through multiple storage mediums, and may reside in a single database or multiple databases (or other data storage techniques). Although the steps, operations, or computations may be presented in a specific order, this order may be changed in different embodiments. In some embodiments, to the extent multiple steps are shown as sequential in this specification, some combination of such steps in alternative embodiments may be performed at the same time. The sequence of operations described herein can be interrupted, suspended, or otherwise controlled by another process, such as an operating system, kernel, etc. The routines can operate in an operating system environment or as stand-alone routines. Functions, routines, methods, steps, and operations described herein can be performed in hardware, software, firmware, or any combination thereof.

[0186] Embodiments described herein can be implemented in the form of control logic in software or hardware or a combination of both. The control logic may be stored in an information storage medium, such as a computer-readable medium, as a plurality of instructions adapted to direct an information processing device to perform a set of steps disclosed in the various embodiments. Based on the disclosure and teachings provided herein, a person of ordinary skill in the art will appreciate other ways and/or methods to implement the invention.

[0187] It is also within the spirit and scope of the invention to implement in software programming or code any of the steps, operations, methods, routines, or portions thereof described herein, where such software programming or code can be stored in a computer-readable medium and can be operated on by a processor to permit a computer to perform any of the steps, operations, methods, routines, or portions thereof described herein. The invention may be implemented by using software programming or code in one or more general purpose digital computers, by using

application specific integrated circuits, programmable logic devices, field programmable gate arrays, optical, chemical, biological, quantum or nanoengineered systems, components and mechanisms may be used. In general, the functions of the invention can be achieved by any means as is known in the art. For example, distributed or networked systems, components and circuits can be used. In another example, communication or transfer (or otherwise moving from one place to another) of data may be wired, wireless, or by any other means.

[0188] A "computer-readable medium" may be any medium that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, system, or device. The computer readable medium can be, by way of example only but not by limitation, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, system, device, propagation medium, or computer memory. Such computer-readable medium shall generally be machine readable and include software programming or code that can be human readable (e.g., source code) or machine readable (e.g., object code). Examples of non-transitory computer-readable media can include random access memories, read-only memories, HDs, data cartridges, magnetic tapes, floppy diskettes, flash memory drives, optical data storage devices, CD-ROMs, and other appropriate computer memories and data storage devices. In an illustrative embodiment, some or all of the software components may reside on a single server computer or on any combination of separate server computers. As one skilled in the art can appreciate, a computer program product implementing an embodiment disclosed herein may comprise one or more non-transitory computer readable media storing computer instructions translatable by one or more processors in a computing environment.

[0189] A "processor" includes any hardware system, mechanism or component that processes data, signals, or other information. A processor can include a system with a general-purpose CPU, multiple processing units, dedicated circuitry for achieving functionality, or other systems. Processing need not be limited to a geographic location, or have temporal limitations. For example, a processor can perform its functions in "real-time," "offline," in a "batch mode," etc. Portions of processing can be performed at different times and at different locations, by different (or the same) processing systems.

[0190] It will also be appreciated that one or more of the elements depicted in the drawings/figures can also be implemented in a more separated or integrated manner, or even removed or rendered as inoperable in certain cases, as is useful in accordance with a particular application. Additionally, any signal arrows in the drawings/Figures should be considered only as exemplary, and not limiting, unless otherwise specifically noted.

[0191] As used herein, the terms "comprises," "comprising," "includes," "including," "has," "having," or any other variation thereof, are intended to cover a non-exclusive inclusion. For example, a process, product, article, or apparatus that comprises a list of elements is not necessarily limited only those elements but may include other elements not expressly listed or inherent to such process, product, article, or apparatus.

[0192] Furthermore, the term "or" as used herein is generally intended to mean "and/or" unless otherwise indicated. For example, a condition A or B is satisfied by any one of the following: A is true (or present) and B is false (or not present), A is false (or not present) and B is true (or present), and both A and B are true (or present). As used herein, that follow, a term preceded by "a set", "a" or "an" (and "the" when antecedent basis is "a" or "an") includes both singular and plural of such term, unless clearly indicated otherwise (i.e., that the reference "a set", "a" or "an" clearly indicates only the singular or only the plural). Also, as used in the description herein the meaning of "in" includes "in" and "on" unless the context clearly dictates otherwise.

[0193] Although the foregoing specification describes specific embodiments, numerous changes in the details of the embodiments disclosed herein and additional embodiments will be apparent to, and may be made by, persons of ordinary skill in the art having reference to this disclosure. In this context, the specification and figures are to be regarded in an illustrative rather than a restrictive

sense, and all such modifications are intended to be included within the scope of this disclosure.

[0194] Embodiments may also be better understood with reference to the included Appendix. It should be understood that this Appendix refers to specific embodiment and that any such restrictive language in the Appendix (e.g., must, should, required, necessary, etc.) should be taken to apply to those specific embodiments and not to embodiment generally.

Claims

1. A system, comprising: a processor; a data store, having email data representing an email conversation comprising a set of emails; and a non-transitory computer readable medium instructions for: receiving one or more scoping criteria; determining a scoped set of emails from the set of emails of the mail conversation based on the one or more scoping criteria; determining a final set of emails based on the scoped set of emails by evaluating the scoped set of emails to eliminate any of the scoped set of emails that are subsumed by another one of the set of scoped emails; and presenting the final set of emails for the email conversation to a user.

2. The system of claim 1, wherein the email data comprises an email graph representing the set of emails, the email graph comprising a set of nodes, wherein each node represents an email of the email conversation and each node comprises a subsumed by field identifying any of the set of emails of the email conversation that include the content of the email represented by that node.

3. The system of claim 2, wherein the content of the email comprise text of the email and any attachments of the email.

4. The system of claim 2, wherein the email graph comprises a raw graph and a simplified graph.

5. The system of claim 2, wherein the subsumed by field of each identifying any of the set of emails of the email conversation that include the content of the email represented by that node, identifies nodes in the email graph representing any of the set of emails of the email conversation that include the content of the email represented by that node.

6. The system of claim 1, wherein the scoping criteria include a search term or a data range.

7. A method, comprising: receiving one or more scoping criteria; determining, from email data representing an email conversation comprising a set of emails, a scoped set of emails from the set of emails of the mail conversation based on the one or more scoping criteria; determining a final set of emails based on the scoped set of emails by evaluating the scoped set of emails to eliminate any of the scoped set of emails that are subsumed by another one of the set of scoped emails; and presenting the final set of emails for the email conversation to a user.

8. The method of claim 7, wherein the email data comprises an email graph representing the set of emails, the email graph comprising a set of nodes, wherein each node represents an email of the email conversation and each node comprises a subsumed by field identifying any of the set of emails of the email conversation that include the content of the email represented by that node.

9. The method of claim 8, wherein the content of the email comprise text of the email and any attachments of the email.

10. The method of claim 8, wherein the email graph comprises a raw graph and a simplified graph.

11. The method of claim 8, wherein the subsumed by field of each identifying any of the set of emails of the email conversation that include the content of the email represented by that node, identifies nodes in the email graph representing any of the set of emails of the email conversation that include the content of the email represented by that node.

12. The method of claim 7, wherein the scoping criteria include a search term or a data range.

13. A non-transitory computer readable medium, comprising instructions for: receiving one or more scoping criteria; determining, from email data representing an email conversation comprising a set of emails, a scoped set of emails from the set of emails of the mail conversation based on the one or more scoping criteria; determining a final set of emails based on the scoped set of emails by evaluating the scoped set of emails to eliminate any of the scoped set of emails that are subsumed

by another one of the set of scoped emails; and presenting the final set of emails for the email conversation to a user.

14. The non-transitory computer readable medium of claim 13, wherein the email data comprises an email graph representing the set of emails, the email graph comprising a set of nodes, wherein each node represents an email of the email conversation and each node comprises a subsumed by field identifying any of the set of emails of the email conversation that include the content of the email represented by that node.

15. The non-transitory computer readable medium of claim 14, wherein the content of the email comprise text of the email and any attachments of the email.

16. The non-transitory computer readable medium of claim 14, wherein the email graph comprises a raw graph and a simplified graph.

17. The non-transitory computer readable medium of claim 14, wherein the subsumed by field of each identifying any of the set of emails of the email conversation that include the content of the email represented by that node, identifies nodes in the email graph representing any of the set of emails of the email conversation that include the content of the email represented by that node.

18. The non-transitory computer readable medium of claim 13, wherein the scoping criteria include a search term or a data range.