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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 August 2021 (12.08.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/156684 A1

(51) International Patent Classification:
G06F 16/36 (2019.01)

(21) International Application Number:
PCT/IB2021/050116

(22) International Filing Date:
08 January 2021 (08.01.2021)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
16/785,329 07 February 2020 (07.02.2020) US

(71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).

(71) Applicants *(for MG only)*: **IBM UNITED KINGDOM LIMITED** [GB/GB]; PO Box 41, North Harbour, Portsmouth, Hampshire PO6 3AU (GB). **IBM (CHINA) INVESTMENT COMPANY LIMITED** [CN/CN]; 25/F, Pangu Plaza, No. 27, Central North 4th Ring Road, Chaoyang District, Beijing 100101 (CN).

(72) Inventors: **SIRANGIMOORTHY, Jothilakshmi**; IBM Corporation, 100 Phoenix Drive, Ann Arbor, Michigan 48108-2600 (US). **RITWIK, Ray**; IBM Corporation, 4205 South Miami Boulevard, Durham, NC 27703-9141 (US). **WANG, Hui**; IBM Corporation, 100 Phoenix Drive, Ann Arbor, Michigan 48108-2600 (US). **RAND, Jonathan**; IBM Corporation, 100 Phoenix Drive, Ann Arbor, Michigan 48108-2600 (US). **CARRIER, Scott**; IBM Corporation, Intellectual Property Law, 4205 South Miami Boulevard, Durham, North Carolina 27703-9141 (US).

(74) Agent: **FOURNIER, Kevin**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Hursley, Winchester, Hampshire SO21 2JN (GB).

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

(54) Title: EXTRACTING INFORMATION FROM UNSTRUCTURED DOCUMENTS USING NATURAL LANGUAGE PROCESSING AND CONVERSION OF UNSTRUCTURED DOCUMENTS INTO STRUCTURED DOCUMENTS

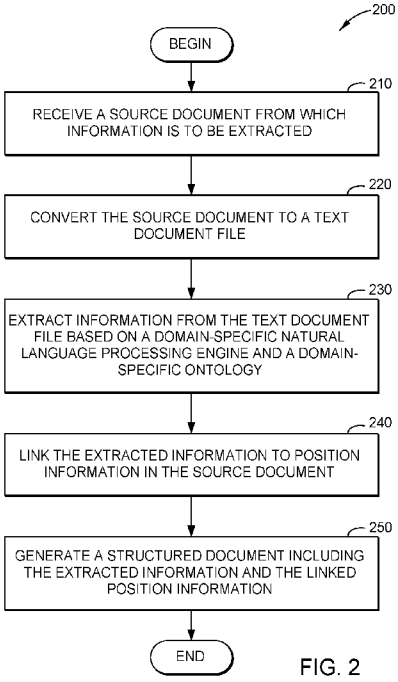


FIG. 2

(57) Abstract: A method for extracting information from textual content, related to the field of machine learning model generation. The method generally includes receiving a training data set including a plurality of documents having related textual strings. A relevancy model is generated from the training data set. The relevancy model is generally configured to generate relevance scores for a plurality of words extracted from the plurality of documents. A knowledge graph model illustrating relationships between the plurality of words extracted from the plurality of documents is generated from the training data set. The relevancy mode-and the knowledge graph model are aggregated into a complimentary model including a plurality of nodes from the knowledge graph model and weights associated with edges between connected nodes, wherein the weights comprise relevance scores generated from the relevancy model, and the complimentary model is deployed for use in analyzing documents.

WO 2021/156684 A1

[Continued on next page]

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

EXTRACTING INFORMATION FROM UNSTRUCTURED DOCUMENTS USING NATURAL LANGUAGE PROCESSING AND CONVERSION OF UNSTRUCTURED DOCUMENTS INTO STRUCTURED DOCUMENTS

BACKGROUND

[0001] Aspects of the present disclosure relate to extracting content from unstructured documents using natural language processing techniques, and more specifically to techniques for generating structured documents from unstructured documents and processing the structured documents using natural language processing techniques.

[0002] Many real-world operations are defined in terms of complex, lengthy documents or sets of documents. These documents may generally be unstructured or semi-structured, in which relevant information is included in paragraphs of textual content or in tables or other organizational constructs that group similar information together. For example, in a healthcare scenario, an employer may offer various health insurance plans to its employers, and the various health insurance plans may be summarized in a summary plan document that identifies, in a table or other organizational construct, various characteristics of the health insurance plans offered to an employee. The summary plan document may further include additional information that may be applicable to all plans offered to an employee or certain plans offered to an employee, and this additional information may be maintained in unstructured text (e.g., paragraphs, bulleted lists, etc.). In another example, retirement plans may also be summarized in summary plan documents that outline relevant information about the various retirement plans offered to an employee in a combination of tabular and unstructured text. In still further examples, various documents in industrial environments may include summaries of requirements for physical goods to be produced, such as size, materials, tolerances, and the like, as well as more detailed information included in unstructured text.

[0003] Generally, the documents discussed above may include a significant amount of domain-specific information. For example, a healthcare summary plan document may include information that is specific to the domain of health insurance plans, such as copay information, in-network/out-of-network definitions, and the like. A retirement summary plan document may include information that is specific to the domain of retirement plans, such as minimum age requirements for participation, vesting information, and the like. While these documents include significant amounts of domain-specific information, these documents may be difficult to interpret and to extract information from.

SUMMARY

[0004] One aspect of the present disclosure provides a method for extracting information from an unstructured document. The method generally includes receiving a source document from which information is to be extracted. The source document is converted to a text document file. Information is extracted from the text document file based on a

domain-specific natural language processing engine and a domain-specific ontology. The extracted information is linked to position information in the source document, and a structured document is generated. The structured document generally includes the extracted information and the linked position information.

[0005] Another aspect of the present disclosure provides a system for extracting information from an unstructured document. The system generally includes a processor and a memory having instructions which, when executed by the processor, performs an operation for extracting information from an unstructured document. The operation generally includes receiving a source document from which information is to be extracted. The source document is converted to a text document file. Information is extracted from the text document file based on a domain-specific natural language processing engine and a domain-specific ontology. The extracted information is linked to position information in the source document, and a structured document is generated. The structured document generally includes the extracted information and the linked position information.

[0006] Still another aspect of the present disclosure provides a computer-readable medium having instructions stored thereon which, when executed by a processor, performs an operation for extracting information from an unstructured document. The operation generally includes receiving a source document from which information is to be extracted. The source document is converted to a text document file. Information is extracted from the text document file based on a domain-specific natural language processing engine and a domain-specific ontology. The extracted information is linked to position information in the source document, and a structured document is generated. The structured document generally includes the extracted information and the linked position information.

[0007] One aspect of the present disclosure provides a method for presenting information extracted from an unstructured document in a graphical user interface. The method generally includes receiving an unstructured document and a structured document including information extracted from the unstructured document and position information associated with the extracted information. The unstructured document is rendered in a first pane, and a graphical rendering of the structured document is rendered in a second pane. The graphical rendering generally may be a structure in which content from the structured document is displayed in a hierarchical format. Each element in the structured document is linked to the rendered unstructured document based on position information included in the structured document.

[0008] Another aspect of the present disclosure provides a system. The system generally includes a processor and a memory having instructions stored thereon which, when executed by the processor, performs an operation for presenting information extracted from an unstructured document in a graphical user interface. The method generally includes receiving an unstructured document and a structured document including information extracted from the unstructured document and position information associated with the extracted information. The unstructured document is rendered in a first pane, and a graphical rendering of the structured document is rendered in a second pane. The

graphical rendering generally may be a structure in which content from the structured document is displayed in a hierarchical format. Each element in the structured document is linked to the rendered unstructured document based on position information included in the structured document.

[0009] Still another aspect of the present disclosure provides a computer-readable medium having instructions stored thereon which, when executed by a processor, performs an operation for presenting information extracted from an unstructured document in a graphical user interface. The operation generally includes receiving an unstructured document and a structured document including information extracted from the unstructured document and position information associated with the extracted information. The unstructured document is rendered in a first pane, and a graphical rendering of the structured document is rendered in a second pane. The graphical rendering generally may be a structure in which content from the structured document is displayed in a hierarchical format. Each element in the structured document is linked to the rendered unstructured document based on position information included in the structured document.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0010] Figure 1 illustrates an example networked computing environment in which information is extracted from an unstructured document using structured documents generated from the unstructured document and natural language processing on the unstructured and structured documents, according to one embodiment.

[0011] Figure 2 illustrates example operations for extracting information from an unstructured document using structured documents and natural language processing, according to one embodiment.

[0012] Figure 3 illustrates example operations for presenting information extracted from an unstructured document in a graphical user interface, according to one embodiment.

[0013] Figure 4 illustrates an example graphical user interface in which an unstructured document and a structured document including information extracted from the unstructured document is displayed, according to one embodiment.

[0014] Figure 5 illustrates an example graphical user interface in which visual effects are rendered in relation to information extracted from an unstructured document, according to one embodiment.

[0015] Figure 6 illustrates an example system in which aspects of the present disclosure may be performed.

DETAILED DESCRIPTION

[0016] Unstructured documents are generally documents in which information is stored in a free-form manner that is not dictated by an underlying structure or relationships between information stored in the document. Unstructured domain-specific documents, such as summary plan description documents, technical specification documents, and the like generally are large documents that are difficult to understand and include a significant amount

of domain-specific information. To extract information from these domain-specific documents, the documents may be converted from an image-based format (e.g., the Portable Document Format) to a plaintext file, and the plaintext file may be processed through a natural language processing engine and/or other analytics engine to extract relevant information from the document. However, conversion of a document to a plaintext file may result in the loss of some information. For example, positional information that may be relevant to content in a document, such as the position of cells in a table and the associated text in each cell, may be lost in the conversion of a document in an image-based format to a plaintext file. Because the positional information and other relevant information may be lost in the conversion of a document from an image-based format to a plaintext file, it may be difficult to map information extracted from the plaintext file back to the original document and to generate a navigable structure that can be used to traverse the document.

[0017] Aspects of the present disclosure provide techniques for extracting information from unstructured documents by generating structured documents that are analyzed by natural language processing engines and other analytics services to extract relevant information. As discussed in further detail below, an unstructured document may be converted to a structured document that maintains contextual information associated with the textual content in the unstructured document. The contextual information can be used, in conjunction with the identification of relevant text by a natural language processing and/or other analytics engines, to generate a structured document that links the extracted relevant text to the position of such text in the unstructured document. Aspects of the present disclosure further provide techniques for rendering and navigating through unstructured documents using a structured document and the information extracted from the structured document. The unstructured document may be rendered beside a navigation structure generated from information extracted from the structured document, and user interaction with the navigation structure may allow for navigation to the location of various elements in the unstructured document. By doing so, a user can quickly navigate to the location of relevant information in the document and can verify that the information was extracted from the unstructured document correctly.

[0018] Figure 1 illustrates an example networked computing environment in which information is extracted from an unstructured document using structured documents generated from the unstructured document, according to an embodiment of the present disclosure. As illustrated, computing environment 100 includes a client device 120, an application server 130, a document analyzer 140, a training data store 150, and an ontology data store 160.

[0019] Client device 120 is generally representative of computing devices on which a user can access a service executing on application server 130 to view and navigate through an unstructured document and access a service executing on document analyzer 140 to initiate extraction of information from an unstructured document. As illustrated, client device includes a software application 122.

[0020] Software application 122 generally provides an interface through which a user can invoke information extraction operations at document analyzer 140 and view the unstructured document through services executing on application server 130. In some embodiments, software application 122 may be a web browser or other application that allows a user to interact with an application hosted remotely on application server 130. Generally, to invoke information extraction operations at document analyzer 140, software application 122 can transmit a command to document analyzer 140 to initiate the process of generating structured documents from the unstructured document and extract information from the structured documents. Generally, the command may include the document to be analyzed or information identifying a location at which the document to be analyzed is stored, and the document may be an unstructured document (e.g., a document in an image-based format) from which information is to be extracted.

[0021] Software application 122 may further provide an interface through which the unstructured document and a navigation pane generated from the analysis of the unstructured document may be viewed. The content of the navigation pane may include one or more links to the locations of extracted information in the unstructured document such that when a user selects a link in the navigation pane, the portion of the unstructured document located at the location associated with the selected link may be viewed in a first pane. Software application 122 may also provide a mechanism through which a user can correct the extracted information displayed in the navigation pane and identify additional relevant information to be included in the navigation pane. The corrected information and additional identified information may be added to a structured document (e.g., stored in document data store 150) specifying the extracted information and location of the extracted information, and the updated structured document may be used in future viewing of the source unstructured document from which the information was extracted.

[0022] Application server 130 generally includes a document viewer 132. Document viewer 132 generally provides an interface through which a user can view an unstructured document and navigate through the unstructured document. The interface may include a first pane in which the unstructured document is rendered and a second pane in which a navigation structure is rendered. The first pane may integrate a document viewer that scrolls to a specified portion of the unstructured document in response to a command to navigate to the specified portion of the document. The command to navigate to the specified portion of the document may include, for example, a command to navigate to an anchor specified in an unstructured document (e.g., a uniform resource locator (URL) including an anchor tag), a command to navigate to a specific page or section of the unstructured document, or the like.

[0023] The navigation structure rendered in the second pane may be, for example, a navigation tree or other hierarchical structure in which links to specified portions of the unstructured document containing the information associated with the link displayed in the navigation tree. As discussed in further detail below, each link in the navigation structure may be associated with a bounding box indicating a location in the unstructured document at which the information associated with the link is found. The bounding box may, for example, be determined by working backwards from one or more intermediate documents generated from the unstructured document to the unstructured

document itself. For example, the bounding box may be generated by first identifying a set of characters in an intermediate plaintext document used by document analyzer 140 to extract information from the unstructured document. The set of characters may be defined, for example, as an offset from a starting character of a text element (e.g., a paragraph, cell in a table, etc.) in an intermediate document, and the identified offset information in one intermediate document may be linked to position information in other intermediate documents generated from the unstructured document. This position information may include, for example, names of cells in a table, named anchors in a document, absolute or relative position information on a page of the unstructured document (e.g., an absolute pixel location, given an assumed resolution of a page in a document, or a relative pixel location as a percentage of the height and width of a page in the document), and the like. Based on the position information in the structured document and the one or more intermediate documents, the location of extracted information may be identified in the unstructured document, and one or more visual effects may be rendered with respect to the location of the extracted information. For example, upon selection of a link in the navigation structure, the content associated with the link may be highlighted using a color unique to the content associated with the link. In another example, a colored box may be drawn around the content associated with the link.

[0024] In some embodiments, the navigation structure may provide a mechanism for a user to correct the content extracted from an unstructured document. As text recognition systems (e.g., optical character recognition systems) may not always accurately convert text in an image to the correct textual content for various reasons (e.g., image quality, document resolution, etc.), the text extracted from the document may not be accurate. Thus, in a verification process, a user may be permitted to examine the document at the location at which extracted information is linked, and if the extracted information is inaccurate, the user can manually overwrite the machine-extracted information with the accurate data. The accurate data may be written to the structured document storing the data extracted from the unstructured document for future use.

[0025] In some embodiments, document analyzer 140 may not extract all of the relevant data from an unstructured document. For example, document analyzer 140 may time out before analyzing the entirety of an unstructured document, or document content may simply be missed during analysis. While a user is viewing the unstructured document, the user may select relevant content in the unstructured document. Document viewer 132 may parse the selected portion of the unstructured document to identify the selected textual content and the positon of the selected portion. Based on the identified portion of the unstructured document, document viewer 132 can add information to the structured document to include one or more references, in one or more structured text elements, to the specified information deemed to be relevant by a user of the document viewer 132. In some embodiments, document viewer 132 may search the one or more intermediate documents to identify other location pointers in the intermediate documents to associate with the identified portion of the unstructured document and the text extracted therefrom and write the identified location pointers to the structured document with an identification of the intermediate document(s) associated with the identified location pointers.

[0026] Document analyzer 140 generally receives an unstructured document for analysis and extracts information from the unstructured document into a structured document using a natural language processing engine and a domain-specific ontology (e.g., one or more of ontologies 162_1-162_n . The domain-specific ontology may be defined *a priori* based on the type of document that is to be analyzed by document analyzer 140. Where document analyzer 140 is configured to extract information from documents in a specific domain (e.g., healthcare summary plan documents, retirement plan documents, etc.), the domain-specific ontology may be hard-coded in document analyzer 140. Where document analyzer 140 is a general purpose document analyzer that can be used to analyze unstructured documents from various domains, document analyzer 140 can determine the domain-specific ontology to use based on user input (e.g., an explicit indication of the type of document provided to document analyzer 140) or based on a natural language analysis of the unstructured document to identify the domain of the document based on key words in the document (e.g., where words like “hospitalization” or “medical” might be associated with a healthcare document domain, while words like “retirement” or “minimum distribution” might be associated with a retirement document domain). As illustrated, document analyzer 140 includes a document converter 142 and a structured document generator 144.

[0027] Document converter 142 generates one or more intermediate documents from an unstructured document received for analysis. To generate the one or more intermediate documents from the unstructured document, document converter 142 can first convert the unstructured document to plaintext file using an optical character recognition engine or other text recognition engines. The optical character recognition engine may be integral to document converter 142 or a remote service that returns a plaintext file in response to a request to generate a plaintext file from an input document. The plaintext file may be the first of a plurality of intermediate documents generated by document converter 142 for use in extracting information from the unstructured document. The output of the optical character recognition engine may be a plaintext file including one or more annotations indicative of a position in which textual content is located in the unstructured document. For example, the optical character recognition engine may generate a plaintext file with one or more text elements, and each text element in the plaintext file may include information identifying where in the unstructured document each text element is located. In some embodiments, the information identifying a location in the unstructured document associated with a text element may be a unique identifier generated by the optical character recognition engine based, for example, formatting information in the unstructured document. For example, new position indicators may be added based on predetermined rules that generate location markers where the typeface or type formatting is indicative of a section break, where a page break exists in a document, and the like.

[0028] The plaintext file resulting from processing the unstructured document through an optical character recognition engine may be a Hypertext Markup Language (HTML) representation of the unstructured document. Each text element in the HTML representation may include position information associated with the location of the text element the unstructured document. This position information may be, for example, a region within the coordinate system of the unstructured document in which the text extracted from the unstructured document was extracted.

Generally, the plaintext file may include structured text including text content from tables or other organizational structures parsed from the unstructured document. The structured text, which may be formatted in a Javascript Object Notation (JSON) format, may retain the tabular format of text extracted from a table by storing column headers, row headers, and cell content for the table. Positional information in the HTML representation may be represented by starting and ending offsets of the extracted text within an HTML string. For example, given an HTML string that may be represented as an array of *i* characters with indices 0-(*i*-1), a string may be represented by a starting and ending index in the array, as positive offsets from the starting character in the string, or as negative offsets from the *i*-1th character in the string.

[0029] The plaintext file may further be processed into a first intermediate document to add one or more text elements to the intermediate document in a structured format. These text elements may include, for example, the text elements not already included in tables in the plaintext document. Text may be concatenated into a series of flattened text elements, and each text element may be associated with a location in an array of text locations for content in the plaintext file. Generally, the position information may include bounding box coordinates in the unstructured document and character offsets of a string in the plaintext file.

[0030] The first intermediate document may be processed to reorganize tabular content stored in a structured text construct. Generally, the structured text may be reorganized to add row headers, section headers, and other information header information included in the document that signifies information about the content in a portion of a document.

[0031] Finally, the first intermediate document may be processed into a second intermediate document that adds flattened text elements to the body cells of a structured text construct including the contents of a table. Section headers, row headers, column headers, and body cell text may be concatenated into a series of flattened text elements. Location data may be stored for each flattened text element generated by concatenating section headers, row headers, column headers, and body cell text into a flattened text element. As discussed above, the position information may include bounding box coordinates in the unstructured document and character offsets of a string in the plaintext file

[0032] Structured document generator 144 uses the intermediate document(s) generated by document converter 142 to generate a structured document including relevant information in the unstructured document. Generally, to identify relevant information in the unstructured document, structured document generator 144 can use a natural language processing engine trained on a domain-specific corpus of information to process the intermediate document(s) generated by document converter 142. For example a structured document generator 144 used to extract information out of healthcare summary plan documents can use a natural language processing engine trained to recognize relevant information from other healthcare plan description documents; a structured document generator 144

used to extract information out of retirement plan documents can use a natural language processing engine trained to recognize relevant information from other retirement plan documents, and the like. The natural language processing engine may be associated with a domain-specific ontology that identifies relationships between specific domain-specific concepts in a document. For example, an ontology for healthcare summary plan descriptions may include relationships between healthcare-specific constructs, such as in-network versus out-of-network coverage, deductibles, co-pays, and the like.

[0033] During the extraction process, structured document generator 144 can record location information associated with the flattened text elements in the intermediate document and save the extracted text and associated location information in a structured document. In some embodiments, the structured document may also include the bounding box associated with the origin of the text in the unstructured document. Once structured document generator 144 generates a structured document according to the domain-specific ontology, structured document generator 144 can commit the structured document to document data store 150. In some embodiments, structured document generator 144 can generate the structured document as a parseable text file (e.g., in a JSON format). In some embodiments, structured document generator 144 can generate the structured document as one or more database records which, when combined, form a structured document that identifies the relevant information extracted from the unstructured document and position information (e.g., text character offsets and bounding boxes in a graphical rendering of the unstructured document) associated with the identified relevant information.

[0034] Document data store 150 is generally representative of a data repository in which unstructured documents, structured documents including information extracted from the unstructured documents, and other textual content used to train the natural language processing engine may be stored. Document data store 150 may include a plurality of separate repositories in which unstructured and structured may be stored. These repositories may be, for example, repositories for different domains of documents, such as technical requirement documents, medical diagnosis documents, and other classes of documents which may be analyzed using document analyzer 140. Document data store 150 may be structured as a relational or a non-relational database, a flat file store, a hierarchical file store, or any other storage system in which documents used to train relevancy and knowledge graph models may be stored.

[0035] Ontology data store 160 is generally representative of a data repository in which domain-specific ontologies may be stored for use by document analyzer 140 in extracting relevant data from an unstructured document. As illustrated, ontology data store 160 may store a plurality of ontologies 162_1-162_n. Each ontology 162 may correspond to a specific domain of documents that may be analyzed by document analyzer 140. For example, one ontology 162 may be associated with healthcare plan summary documents, another ontology 162 may be associated with retirement plan summary documents, and the like.

[0036] Figure 2 illustrates example operations 200 that may be performed by a system to extract information from an unstructured document using natural language processing techniques that preserves positional information associated with the extracted information. As discussed, the preservation of positional information associated with the extracted information may be used in rendering an unstructured document with a navigation structure that allows a user to jump to locations in the unstructured document at which extracted information is stored.

[0037] As illustrated, operations 200 begin at block 210, where a system receives a source document from which information is to be extracted. The source document may be, for example, a document in a file format used by word processing programs (e.g., an OpenDocument Format file, an Office Open XML file, etc.), an image-based file format (e.g., a Portable Document Format file), or the like. Generally, the source document may be associated with a specific domain of knowledge that indicates the types of information that may be relevant to the document (or type of document).

[0038] At block 220, the system converts the source document to a text document file. To convert the source document to a text document file, a first intermediate file may be generated by passing the unstructured source document to an optical character recognition engine. The optical character recognition engine may generate a plaintext file that includes the text content extracted from the source document and a structure defining the characteristics of tabular data included in the source document. The first intermediate file may include position information (e.g., vertical/horizontal position information using the coordinate system of the source document) that a system can use to identify locations of relevant text in the source document. One or more other intermediate files may be generated to convert unstructured text content into structured text content that can be analyzed by a natural language processing engine to extract information from the text document. The resulting file may be a structured text document representative of the unstructured source document that includes position information for each text element included in the source document.

[0039] At block 230, the system extracts information from the text document file based on a domain-specific natural language processing engine and a domain-specific ontology. The domain-specific natural language processing engine may be trained to recognize information of relevance in a particular type of document (e.g., healthcare summary plan descriptions, retirement summary plan descriptions, industrial specification documents, etc.), and the domain-specific ontology may specify relationships between different words or concepts in the particular type of document.

[0040] At block 240, the system links the extracted information to position information in the source document. As discussed, the extracted information may be associated with a particular set of characters in the text document file and/or one or more intermediate documents generated from the source document. The set of characters, which may be identified by a character offset or indices of characters in a document, may be linked to coordinate position information

in the unstructured document through positioning information (e.g., table cell identifiers) included on one or more intermediate documents.

[0041] At block 250, the system generates a structured document including the extracted information and the linked position information. As discussed, the structured document may be a document written according to a hierarchical schema in which information extracted from the document is stored. In some embodiments, the structured document may be stored in a data repository as a series of database records that, when aggregated, may constitute a single structured document that may be used to generate a navigation structure for the document.

[0042] Figure 3 illustrates example operations 300 that may be performed to render an unstructured document and a navigation structure generated from a structured document including information extracted from the unstructured document.

[0043] Operations 300 begin at block 310, where a system receives an unstructured document and an associated structured document including information extracted from the unstructured document and position information associated with the extracted information. The structured document may be formatted using a hierarchical schema that can be translated into a hierarchical navigation structure. Generally, the hierarchical schema may be structured such that nesting of elements within other elements indicates a parent-child relationship between different pieces of data extracted from the structured document.

[0044] At block 320, the system renders the unstructured document in a first pane of a graphical user interface. The first pane may include a document viewer that supports documents in a set of document file formats typically used for unstructured documents. For example, the document viewer may be an HTML rendering engine used to display documents in the HTML file format, a PDF rendering engine, and the like. In some cases, the document viewer may allow users to select information in a document but may not include editing functions that allow a user to change the content of the unstructured document.

[0045] At block 330, the system generates a graphical rendering of the structured document in a second pane of the graphical user interface. The graphical rendering of the structured document may be, for example, a navigation tree or other hierarchical structure that can be used to represent hierarchical data in the structured document (e.g., as a nested list of data). As discussed, the structured document may include extracted information in a hierarchy of textual elements in the structured document, where different levels of information represent parent-child relationships between higher and lower levels of information.

[0046] At block 340, the system links each element in the graphical rendering of the structured document to the unstructured document based on position information included in the structured document. By linking elements in the

graphical rendering of the structured document to the unstructured document, the system can allow for navigation through the unstructured document based on interaction with elements in the graphical rendering of the structured document. As discussed, the position information may be determined based on position information included in one or more intermediate documents generated from the structured document, which may be character offsets relative to a string in an intermediate document. These character offsets may be mapped back to coordinate position information in the unstructured document. Each element in the graphical rendering of the structured document may thus be linked to the coordinates in the unstructured document at which the extracted text is located, and user selection of an element in the graphical rendering of the structured document may cause the document viewer to render the unstructured document at the coordinate position associated with the selected element.

[0047] Figure 4 illustrates a graphical user interface displaying an unstructured document and a navigation pane for the unstructured document populated based on a structured document including information extracted from the unstructured document. As illustrated, graphical user interface 400 includes a first pane 410 and a second pane 420.

[0048] First pane 410 generally includes a document viewer that allows a user to view an unstructured document, and second pane 420 generally includes a hierarchical structure viewer in which the structured document is rendered. Each element from the structured document displayed in the hierarchical structure viewer of second pane 420 may be user-selectable. When a user selects an element 422 in second pane 420, corresponding to an element extracted from the unstructured document rendered in first pane 410, a system can display additional information about the selected element in an information viewer 424 in second pane 420. To do so, the system can retrieve information from the structured document associated with the selected element. The retrieved information may include, for example, positional information associated with the selected element and other relevant information associated with the selected element. For example, as illustrated, the selected element 422 is associated with information located at page 11 of the unstructured document displayed in first pane 410, references a “Lifetime Limit Individual” property of the plan(s) described in the unstructured document, and has a value of “$2,500”. This information may be displayed in information viewer 424 in a read-only format or an editable format, based on user selection(s) in the second pane 420.

[0049] The system may further display the portion of the unstructured document at which the selected element is located in first pane 410. In some embodiments, the system can additionally render one or more visual effects in respect of the portion of the document at which the selected element is located. For example, a bounding box 412 can be drawn in first pane 410 around the portion of the document at which the selected element 422 is located, text in the portion of the document at which the selected element is located may be highlighted, and so on. These visual effects may be rendered based on the positional information associated with the selected element 422 in the structured document generated from the unstructured document displayed in first pane 410. As discussed, this positional information may include a textual offset in one or more intermediate documents and coordinates in the unstructured document displayed in first pane 410.

[0050] Second pane 420 generally also includes one or more selectable icons or buttons that allow a user to edit or delete existing information extracted from the unstructured document into the structured document and/or add information to the structured document from the unstructured document. For example, each item included in the structured document and extracted from the unstructured document may be associated with an edit button 426 and a delete button 428. If a user selects the edit button 426 associated with an item in the structured document, the system can populate information viewer 424 with the information associated with the item in the structured document and make one or more user interface elements in information viewer 424 editable. When a user indicates that editing is complete, the system can write the edited information back to the structured document. If a user selects the delete button 428 associated with an item in the structured document, the system can remove the item from the structured document and re-render the hierarchical structure displayed in second pane 420.

[0051] To add a value, a user can initiate the process by interacting with the add value button 430 displayed in the second pane 420. After initiating the process, the user can select a portion of the unstructured document in first pane 410. The system can identify the portion of the structured document to add information to based on the position of the selected portion of the unstructured document. At the identified portion of the structured document, the system can add one or more text elements representing the selected portion of the unstructured document and position information in the unstructured document such that when rendered in second pane 420, the selected portion of the document can be displayed in first pane 410.

[0052] Figure 5 illustrates a graphical user interface displaying an unstructured document and a navigation pane for the unstructured document. Like Figure 4, graphical user interface 500 includes a first pane 410 and a second pane 420.

[0053] As illustrated, extracted text 528 may be selected for analysis by a user by clicking on a button provided in second pane 420 for viewing extended information (e.g., location and relationship information) about extracted text 528. In some embodiments, by selecting extracted text 528, the system can render one or more visual effects for elements in the structured document that correspond to parent elements of the selected extracted text 528. In this example, extracted text 528 may be an immediate child of extracted text 526, which may be an immediate child of extracted text 524, which may be an immediate child of extracted text 522. Extracted text 522 may represent the ultimate parent text object in the hierarchical navigation structure for extracted text 528. In this example, the user has selected a monetary amount that relates to emergency room visits, which in turn relates to copay amounts the user is responsible for, which ultimately relates to information about a “low deductible PPO” in the unstructured document rendered in first pane 410.

[0054] Based on user selection of extracted text 528, the system can examine positional information in the structured document generated from the unstructured document rendered in first pane 410 to identify the locations of

each of extracted texts 522, 524, 526, and 528 in the unstructured document. As discussed, the identified locations of each of the extracted texts may include text offsets in a string and coordinate positions of the extracted texts in the unstructured document. Based on the identified locations of an extracted text, the system can render one or more visual effects around the relevant text in the unstructured document rendered in first pane 410 and corresponding visual effects around the extracted text in second pane 420. For example, different colored bounding boxes may be rendered in first pane 410 and second pane 420 for each extracted text relevant to the user selected extracted text (i.e., relevant to extracted text 528). As illustrated, bounding box 512 may correspond to extracted text 522; bounding box 514 may correspond to extracted text 524; bounding box 516 may correspond to extracted text 526; and bounding box 518 may correspond to extracted text 528. To allow for quick recognition of extracted text elements in second pane 420 and the corresponding relevant text in the unstructured document rendered in first pane 410, the system can render the same or similar visual effects to the bounding box in first pane 410 and the associated relevant text in second pane 420. In this example, bounding box 512 and a box rendered around the associated extracted text 522 may have a border in a first color; bounding box 512 and a box rendered around the associated extracted text 524 may have a border in a second color; bounding box 516 and a box rendered around the associated extracted text 526 may have a border in a third color; and bounding box 518 and a box rendered around the associated extracted text 528 may have a border in a fourth color. In other examples, different visual effects may be rendered in the unstructured document displayed in first pane 410 for each of the extracted texts 522, 524, 526, and 528. For example, different colored highlighting may be applied to the relevant text in the unstructured document displayed in first pane 410 based on the coordinates of the bounding boxes discussed above; the color of the relevant text in the unstructured document displayed in first pane 410 may be changed; and so on.

[0055] Figure 6 illustrates an example system 600 that extracts information from an unstructured document into a structured document that can be used to navigate through the unstructured document using natural language processing techniques and an ontology associated with a type of the unstructured document, according to an embodiment. As illustrated, system 600 includes, without limitation, a central processing unit 602, one or more I/O device interfaces 604, which may allow for the connection of various I/O devices 614 (e.g., keyboards, displays, mouse devices, pen input, etc.) to the system 600, network interface 606, a memory 608, storage 610, and an interconnect 612.

[0056] CPU 602 may retrieve and execute programming instructions stored in the memory 608. Similarly, the CPU 602 may retrieve and store application residing in the memory 608. The interconnect 612 transmits programming instructions and application data among the CPU 602, I/O device interface 604, network interface 606, memory 608, and storage 610. CPU 602 is included to be representative of a single CPU, multiple CPUs, a single CPU having multiple processing cores, and the like. Additionally, the memory 608 is included to be representative of a random access memory. Furthermore, storage 610 may be a disk drive, a solid state drive, a phase change memory device, or the like. Although shown as a single unit, the storage 610 may be a combination of fixed and/or removable storage

devices, such as fixed disc drives, solid state drives, removable memory cards or optical storage, network attached storage (NAS), or a storage area network (SAN).

[0057] As illustrated, memory 608 includes a document converter 620, a structured document generator 630, and a document viewer 640. Document converter 620 generally receives a document for analysis and generates one or more intermediate plaintext documents that structured document generator 630 can use to generate the structured document. The received document may be, for example, in an image-based file format (e.g., the Portable Document Format) or other document file formats that may be rendered on a screen. Using an optical character recognition engine, document converter 620 can convert the unstructured document to a plaintext document that may serve as the first of a plurality of intermediate documents used to generate the structured document. Text content in tables may be formatted in the first intermediate document in a manner that preserves information about each cell of the table, and successive intermediate documents can add structure around text elements in the unstructured document that are not included in a table in the unstructured document. Each text element may also include position information, such as character offsets and bounding box coordinates in the unstructured document, which may be used by document viewer 640 in rendering the unstructured document and applying visual effects to portions of the unstructured document.

[0058] Structured document generator 630 generally uses the intermediate documents to generate a structured document including relevant information extracted from the unstructured document. To generate a structured document, structured document generator 630 uses a domain-specific natural language processing engine and a domain-specific ontology trained to recognize words or phrases of particular significance to a class of documents. The structured document generated by structured document generator 630 may be a hierarchical document which document viewer 640 can use in rendering a navigation structure that preserves the hierarchy and the relationships between the data extracted from the document.

[0059] Document viewer 640 generally provides a graphical user interface in which an unstructured document is rendered in a first pane of the graphical user interface and a navigation structure generated from a structured document including information extracted from the unstructured document is displayed in a second pane. Generally, when a user interacts with elements in the navigation structure, document viewer 640 can jump to a portion of the unstructured document including the selected text element and apply one or more visual effects to the portion of the unstructured document including the selected text element. Document viewer 640 additionally may allow a user to edit and remove exiting content from a structured document associated with the unstructured document displayed in the first pane and add information from the unstructured document to the structured document.

[0060] Storage 610, as illustrated, includes a document data store 650 and an ontology data store 660. Document data store 650 generally serves as a repository in which structured documents including information extracted from one or more unstructured documents may be stored. Ontology data store 660 generally includes a priori

defined relationships between data in various domains of documents that may be processed by the system 600. Document data store 650 and ontology data store 660 may be a relational database, a non-relational database, a flat file store, or any other data repository in which encrypted data may be stored and retrieved.

[0061] The descriptions of the various embodiments of the present invention have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments. The terminology used herein was chosen to best explain the principles of the embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments disclosed herein.

[0062] In the following, reference is made to embodiments presented in this disclosure. However, the scope of the present disclosure is not limited to specific described embodiments. Instead, any combination of the following features and elements, whether related to different embodiments or not, is contemplated to implement and practice contemplated embodiments. Furthermore, although embodiments disclosed herein may achieve advantages over other possible solutions or over the prior art, whether or not a particular advantage is achieved by a given embodiment is not limiting of the scope of the present disclosure. Thus, the following aspects, features, embodiments and advantages are merely illustrative and are not considered elements or limitations of the appended claims except where explicitly recited in a claim(s). Likewise, reference to "the invention" shall not be construed as a generalization of any inventive subject matter disclosed herein and shall not be considered to be an element or limitation of the appended claims except where explicitly recited in a claim(s).

[0063] Aspects of the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system."

[0064] The present invention may be a system, a method, and/or a computer program product. The computer program product may include a computer readable storage medium (or media) having computer readable program instructions thereon for causing a processor to carry out aspects of the present invention.

[0065] The computer readable storage medium can be a tangible device that can retain and store instructions for use by an instruction execution device. The computer readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. A non-exhaustive list of more specific examples of the computer readable storage medium includes the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only

memory (EPROM or Flash memory), a static random access memory (SRAM), a portable compact disc read-only memory (CD-ROM), a digital versatile disk (DVD), a memory stick, a floppy disk, a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon, and any suitable combination of the foregoing. A computer readable storage medium, as used herein, is not to be construed as being transitory signals *per se*, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or other transmission media (e.g., light pulses passing through a fiber-optic cable), or electrical signals transmitted through a wire.

[0066] Computer readable program instructions described herein can be downloaded to respective computing/processing devices from a computer readable storage medium or to an external computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network. The network may comprise copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers and/or edge servers. A network adapter card or network interface in each computing/processing device receives computer readable program instructions from the network and forwards the computer readable program instructions for storage in a computer readable storage medium within the respective computing/processing device.

[0067] Computer readable program instructions for carrying out operations of the present invention may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, or either source code or object code written in any combination of one or more programming languages, including an object oriented programming language such as Smalltalk, C++ or the like, and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider). In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects of the present invention.

[0068] Aspects of the present invention are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems), and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of

blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer readable program instructions.

[0069] These computer readable program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks. These computer readable program instructions may also be stored in a computer readable storage medium that can direct a computer, a programmable data processing apparatus, and/or other devices to function in a particular manner, such that the computer readable storage medium having instructions stored therein comprises an article of manufacture including instructions which implement aspects of the function/act specified in the flowchart and/or block diagram block or blocks.

[0070] The computer readable program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other device to cause a series of operational steps to be performed on the computer, other programmable apparatus or other device to produce a computer implemented process, such that the instructions which execute on the computer, other programmable apparatus, or other device implement the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0071] The flowchart and block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of instructions, which comprises one or more executable instructions for implementing the specified logical function(s). In some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts or carry out combinations of special purpose hardware and computer instructions.

[0072] While the foregoing is directed to embodiments of the present invention, other and further embodiments of the invention may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.

CLAIMS

1. A method for extracting information from an unstructured document, comprising:

receiving a source document from which information is to be extracted;

converting the source document to a text document file;

extracting information from the text document file based on a domain-specific natural language processing engine and a domain-specific ontology;

linking the extracted information to position information in the source document; and

generating a structured document including the extracted information and the linked position information.

2. The method of claim 1, wherein:

the source document includes a plurality of text elements, and

converting the source document comprises:

generating a position indicator for a plurality of text elements in the source document, and

generating a first intermediate text document including content of each of the plurality of text elements and an associated position indicator for each of the plurality of text elements.

3. The method of claim 2, wherein the generated first intermediate text document comprises a structured document formatted according to a predefined schema.

4. The method of claim 2, wherein:

the plurality of text elements comprise cells in a table, and

the first intermediate text document replaces the table with a structured representation, with each cell in the table being represented by a separate set of elements in the structured representation and being associated with a position indicator in the source document.

5. The method of claim 4, further comprising:

generating a second intermediate text document from the first intermediate text document by concatenating textual content not included in the table into one or more flat text elements, each of the one or more flat text elements including an array of locations representation a location of each respective flat text element in the source document.

6. The method of claim 4, wherein the one or more flat text elements comprise a structured representation of a non-tabular element in the source document.

7. The method of claim 4, further comprising:

concatenating table header and content of a table cell into a flattened text element, the flattened text element including a bounding box defining a position of the table cell in the source document and beginning and ending offsets in the flattened text element.

8. The method of claim 4, wherein extracting information from the source document comprises:

identifying relevant textual information in the text document file and positional offset information associated with the identified relevant textual information; and

generating a bounding box in the source document based on the positional offset information.

9. A system, comprising:

a processor; and

a memory having instructions stored thereon which, when executed by a processor, performs an operation for extracting information from an unstructured document, the operation comprising:

receiving a source document from which information is to be extracted;

converting the source document to a text document file;

extracting information from the text document file based on a domain-specific natural language processing engine and a domain-specific ontology;

linking the extracted information to position information in the source document; and

generating a structured document including the extracted information and the linked position information.

10. The system of claim 9, wherein:

the source document includes a plurality of text elements, and

converting the source document comprises:

generating a position indicator for a plurality of text elements in the source document, and

generating a first intermediate text document including content of each of the plurality of text elements and an associated position indicator for each of the plurality of text elements.

11. The system of claim 10, wherein the generated first intermediate text document comprises a structured document formatted according to a predefined schema.

12. The system of claim 10, wherein:

the plurality of text elements comprise cells in a table, and

the first intermediate text document replaces the table with a structured representation, with each cell in the table being represented by a separate set of elements in the structured representation and being associated with a position indicator in the source document.

13. The method of claim 12, wherein the operation further comprises:

generating a second intermediate text document from the first intermediate text document by concatenating textual content not included in the table into one or more flat text elements, each of the one or more flat text elements including an array of locations representation a location of each respective flat text element in the source document.

14. The method of claim 12, wherein the one or more flat text elements comprise a structured representation of a non-tabular element in the source document.

15. The method of claim 12, wherein the operation further comprises:

concatenating table header and content of a table cell into a flattened text element, the flattened text element including a bounding box defining a position of the table cell in the source document and beginning and ending offsets in the flattened text element.

16. The method of claim 12, wherein extracting information from the source document comprises:

identifying relevant textual information in the text document file and positional offset information associated with the identified relevant textual information; and

generating a bounding box in the source document based on the positional offset information.

17. A computer-readable medium having instructions stored thereon which, when executed by a processor, performs an operation for extracting information from an unstructured document, comprising:

receiving a source document from which information is to be extracted;

converting the source document to a text document file;

extracting information from the text document file based on a domain-specific natural language processing engine and a domain-specific ontology;

linking the extracted information to position information in the source document; and

generating a structured document including the extracted information and the linked position information.

18. The computer-readable medium of claim 17, wherein:

the source document includes a plurality of text elements, and

converting the source document comprises:

generating a position indicator for a plurality of text elements in the source document, and

generating a first intermediate text document including content of each of the plurality of text elements and an associated position indicator for each of the plurality of text elements.

19. The computer-readable medium of claim 18, wherein:

the plurality of text elements comprise cells in a table, and

the first intermediate text document replaces the table with a structured representation, with each cell in the table being represented by a separate set of elements in the structured representation and being associated with a position indicator in the source document.

20. The computer-readable medium of claim 19, further comprising:

generating a second intermediate text document from the first intermediate text document by concatenating textual content not included in the table into one or more flat text elements, each of the one or more flat text elements including an array of locations representation a location of each respective flat text element in the source document.

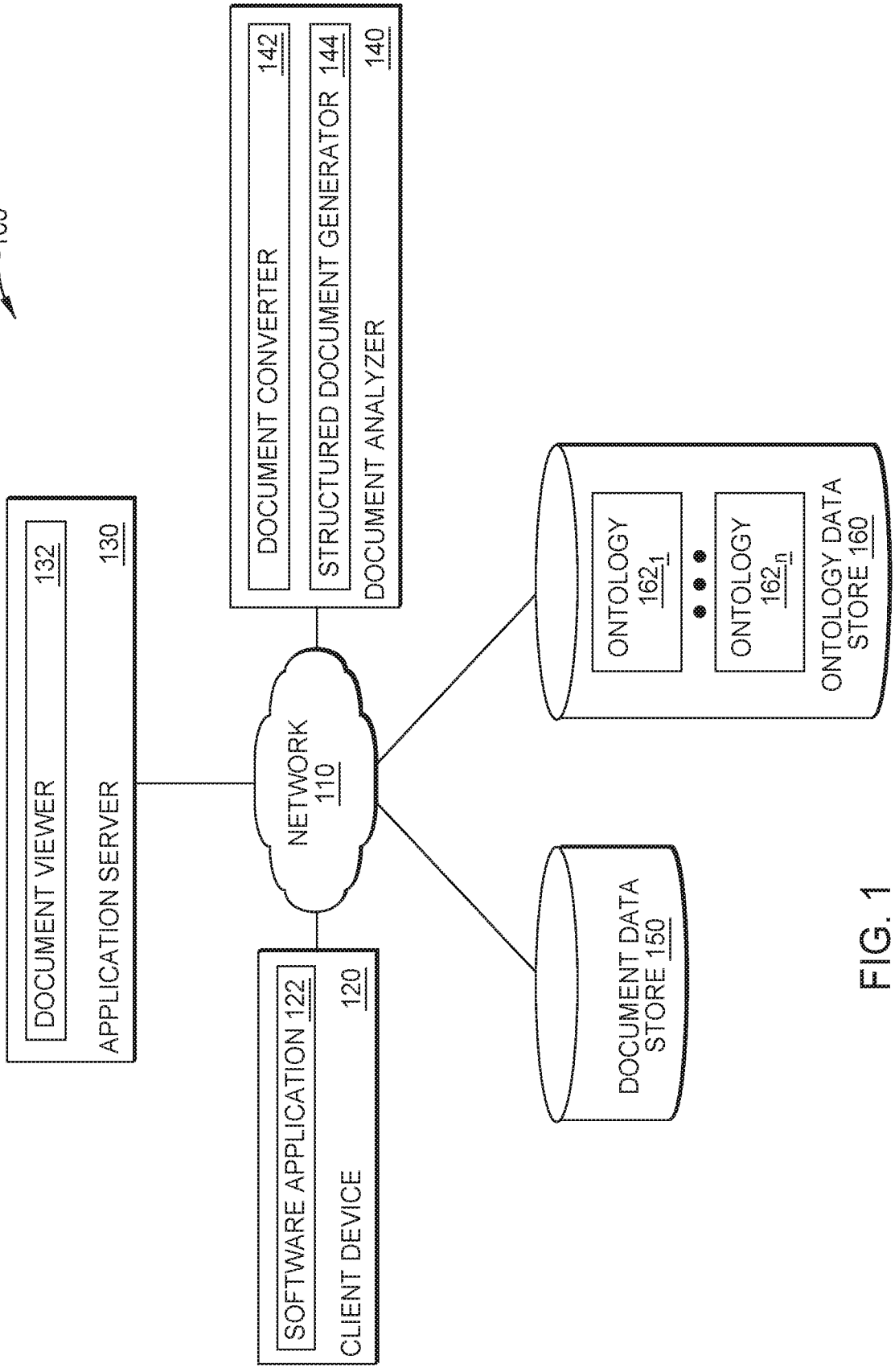

100
DOCUMENT VIEWER 132
APPLICATION SERVER 130
NETWORK 110
DOCUMENT CONVERTER 142
STRUCTURED DOCUMENT GENERATOR 144
DOCUMENT ANALYZER 140
SOFTWARE APPLICATION 122
CLIENT DEVICE 120
ONTOLOGY 162_1
ONTOLOGY 162_n
ONTOLOGY DATA STORE 160
DOCUMENT DATA STORE 150

FIG. 1

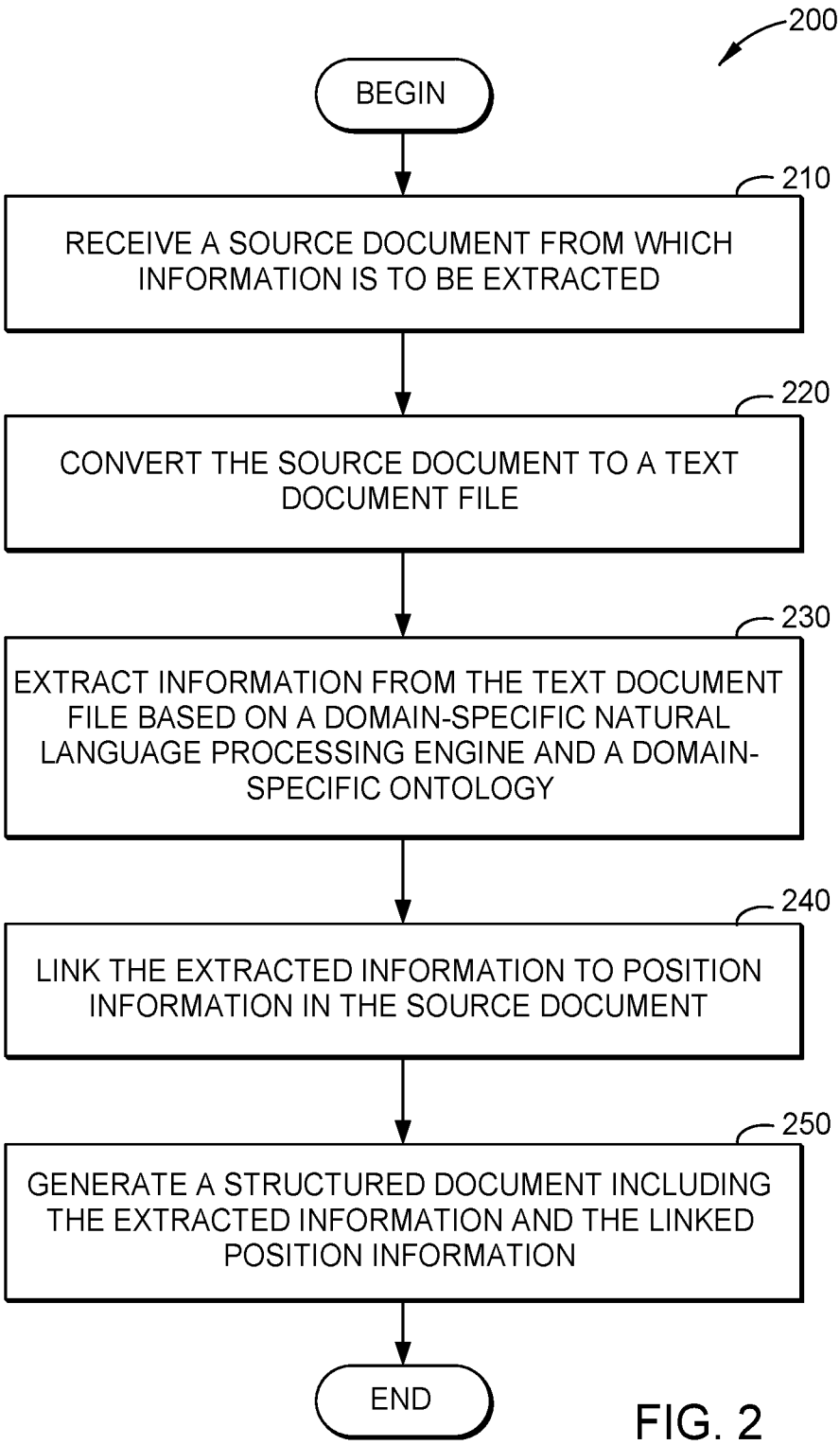

200
BEGIN
210
RECEIVE A SOURCE DOCUMENT FROM WHICH INFORMATION IS TO BE EXTRACTED
220
CONVERT THE SOURCE DOCUMENT TO A TEXT DOCUMENT FILE
230
EXTRACT INFORMATION FROM THE TEXT DOCUMENT FILE BASED ON A DOMAIN-SPECIFIC NATURAL LANGUAGE PROCESSING ENGINE AND A DOMAIN-SPECIFIC ONTOLOGY
240
LINK THE EXTRACTED INFORMATION TO POSITION INFORMATION IN THE SOURCE DOCUMENT
250
GENERATE A STRUCTURED DOCUMENT INCLUDING THE EXTRACTED INFORMATION AND THE LINKED POSITION INFORMATION
END

FIG. 2

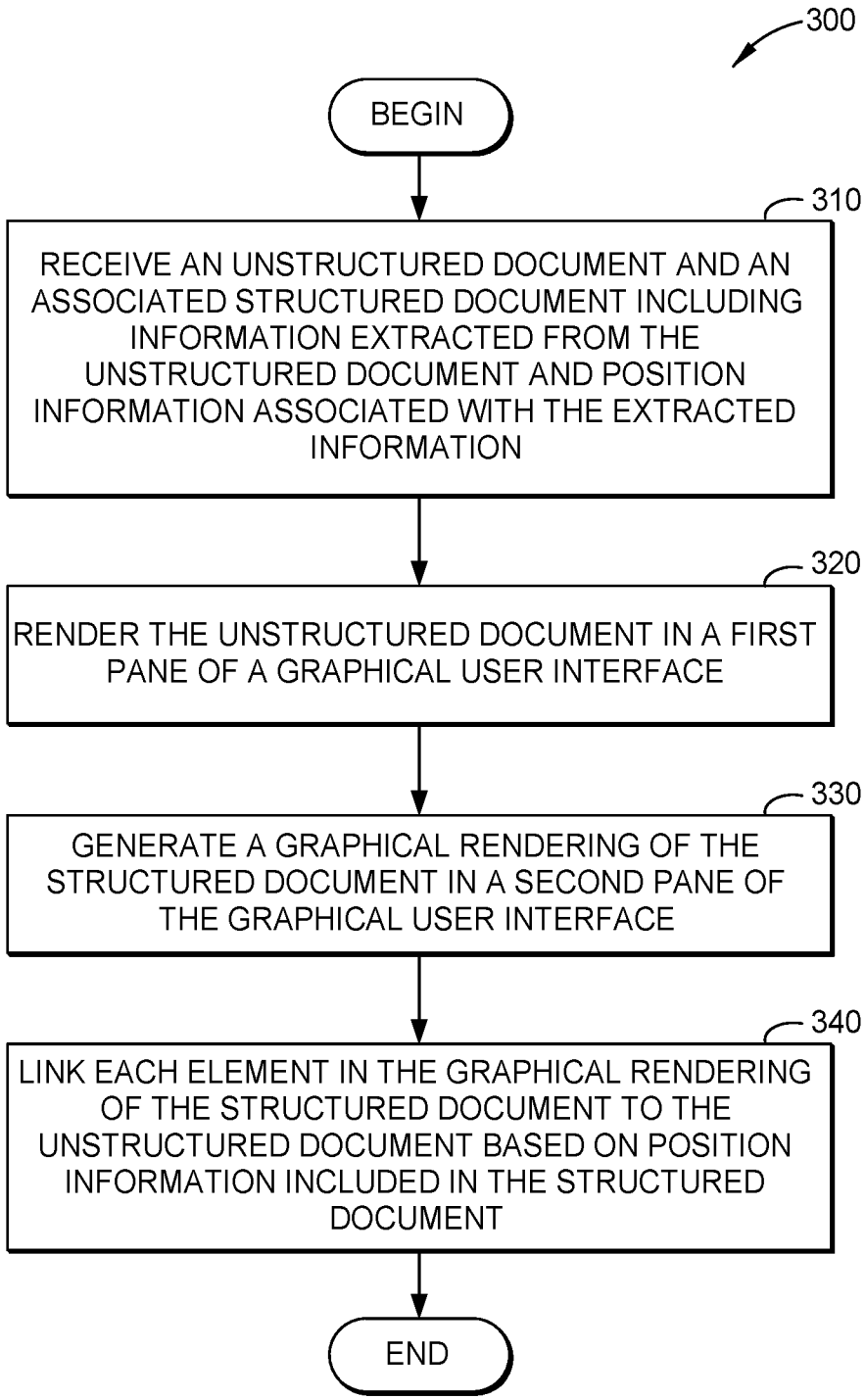

300
BEGIN
310
RECEIVE AN UNSTRUCTURED DOCUMENT AND AN ASSOCIATED STRUCTURED DOCUMENT INCLUDING INFORMATION EXTRACTED FROM THE UNSTRUCTURED DOCUMENT AND POSITION INFORMATION ASSOCIATED WITH THE EXTRACTED INFORMATION
320
RENDER THE UNSTRUCTURED DOCUMENT IN A FIRST PANE OF A GRAPHICAL USER INTERFACE
330
GENERATE A GRAPHICAL RENDERING OF THE STRUCTURED DOCUMENT IN A SECOND PANE OF THE GRAPHICAL USER INTERFACE
340
LINK EACH ELEMENT IN THE GRAPHICAL RENDERING OF THE STRUCTURED DOCUMENT TO THE UNSTRUCTURED DOCUMENT BASED ON POSITION INFORMATION INCLUDED IN THE STRUCTURED DOCUMENT
END

FIG. 3

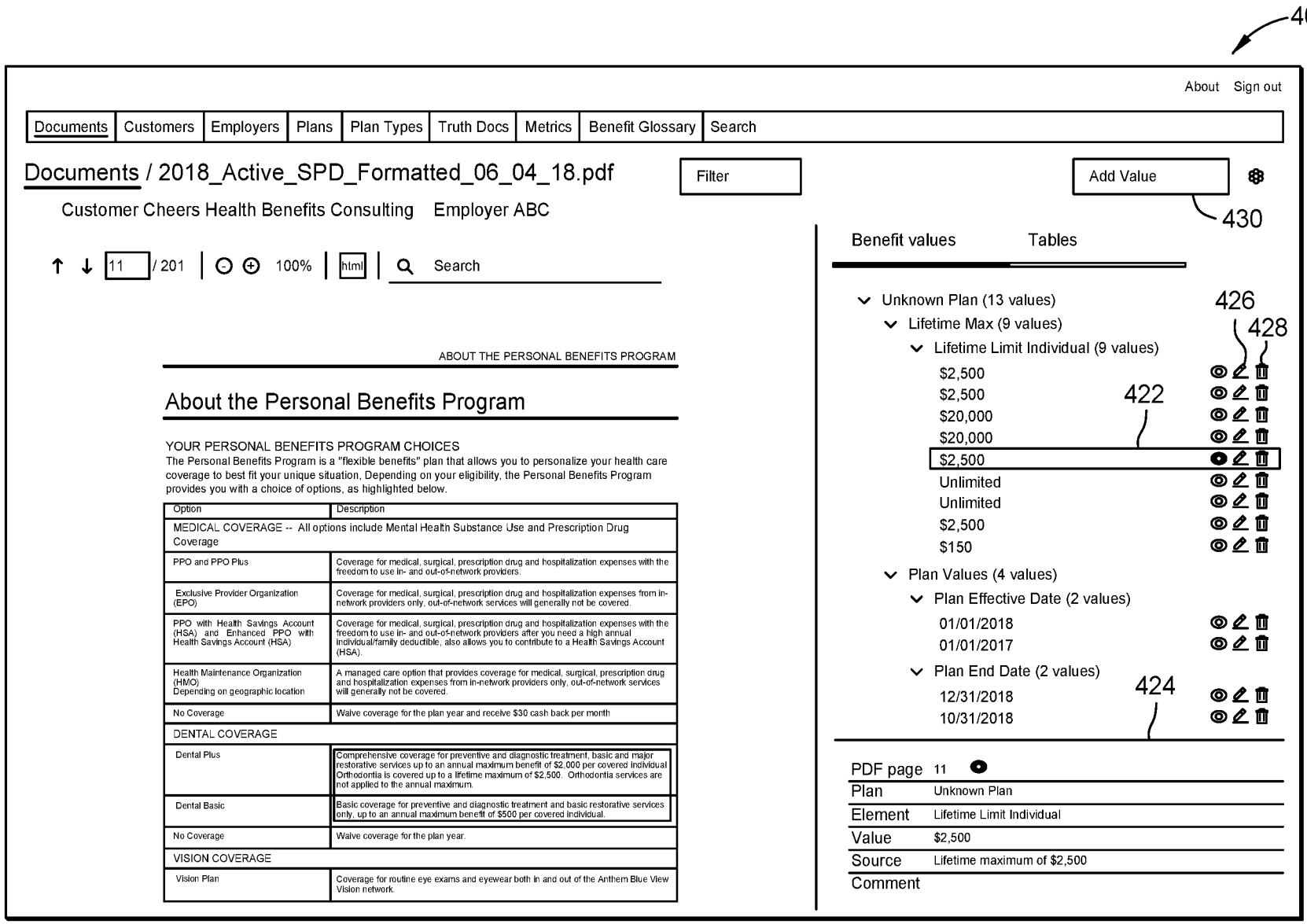

400
About
Sign out
Documents
Customers
Employers
Plans
Plan Types
Truth Docs
Metrics
Benefit Glossary
Search
Documents / 2018_Active_SPD_Formatted_06_04_18.pdf
Filter
Add Value
430
Customer Cheers Health Benefits Consulting
Employer ABC
11
/ 201
100%
html
Search
ABOUT THE PERSONAL BENEFITS PROGRAM
About the Personal Benefits Program
YOUR PERSONAL BENEFITS PROGRAM CHOICES
The Personal Benefits Program is a "flexible benefits" plan that allows you to personalize your health care coverage to best fit your unique situation, Depending on your eligibility, the Personal Benefits Program provides you with a choice of options, as highlighted below.
Option
Description
MEDICAL COVERAGE -- All options include Mental Health Substance Use and Prescription Drug Coverage
PPO and PPO Plus
Coverage for medical, surgical, prescription drug and hospitalization expenses with the freedom to use in- and out-of-network providers.
Exclusive Provider Organization (EPO)
Coverage for medical, surgical, prescription drug and hospitalization expenses from in-network providers only; out-of-network services will generally not be covered.
PPO with Health Savings Account (HSA) and Enhanced PPO with Health Savings Account (HSA)
Coverage for medical, surgical, prescription drug and hospitalization expenses with the freedom to use in- and out-of-network providers after you need a high annual individual/family deductible, also allows you to contribute to a Health Savings Account (HSA).
Health Maintenance Organization (HMO) Depending on geographic location
A managed care option that provides coverage for medical, surgical, prescription drug and hospitalization expenses from in-network providers only; out-of-network services will generally not be covered.
No Coverage
Waive coverage for the plan year and receive $30 cash back per month
DENTAL COVERAGE
Dental Plus
Comprehensive coverage for preventive and diagnostic treatment, basic and major restorative services up to an annual maximum benefit of $2,000 per covered individual Orthodontia is covered up to a lifetime maximum of $2,500. Orthodontia services are not applied to the annual maximum.
Dental Basic
Basic coverage for preventive and diagnostic treatment and basic restorative services only, up to an annual maximum benefit of $500 per covered individual.
No Coverage
Waive coverage for the plan year.
VISION COVERAGE
Vision Plan
Coverage for routine eye exams and eyewear both in and out of the Anthem Blue View Vision network.
Benefit values
Tables
426
428
422
Unknown Plan (13 values)
Lifetime Max (9 values)
Lifetime Limit Individual (9 values)
$2,500
$2,500
$20,000
$20,000
$2,500
Unlimited
Unlimited
$2,500
$150
Plan Values (4 values)
Plan Effective Date (2 values)
01/01/2018
01/01/2017
Plan End Date (2 values)
424
12/31/2018
10/31/2018
PDF page 11
Plan
Unknown Plan
Element
Lifetime Limit Individual
Value
$2,500
Source
Lifetime maximum of $2,500
Comment

FIG. 4

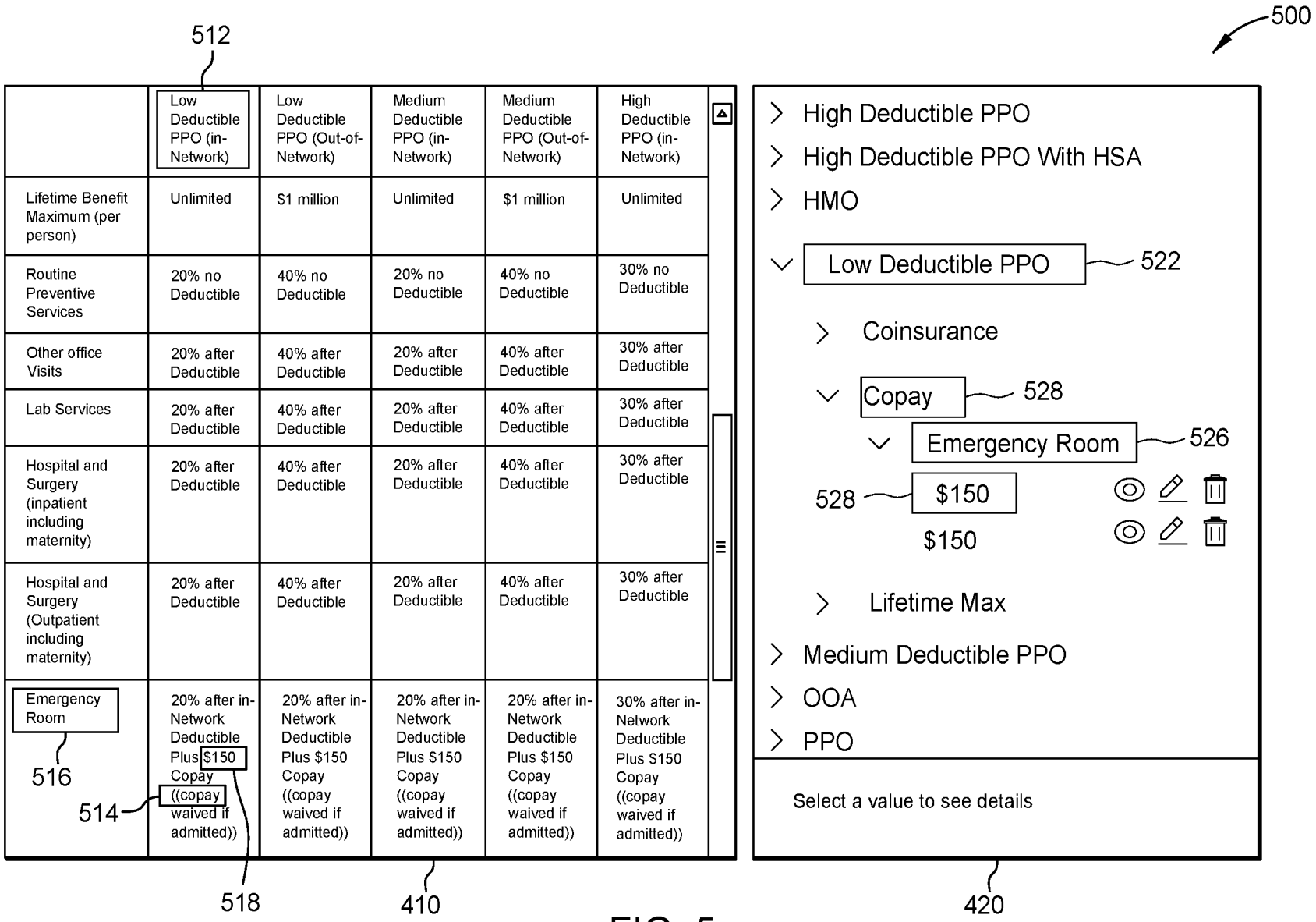

512
500
Low Deductible PPO (in-Network)
Low Deductible PPO (Out-of-Network)
Medium Deductible PPO (in-Network)
Medium Deductible PPO (Out-of-Network)
High Deductible PPO (in-Network)
Lifetime Benefit Maximum (per person)
Unlimited
$1 million
Unlimited
$1 million
Unlimited
Routine Preventive Services
20% no Deductible
40% no Deductible
20% no Deductible
40% no Deductible
30% no Deductible
Other office Visits
20% after Deductible
40% after Deductible
20% after Deductible
40% after Deductible
30% after Deductible
Lab Services
20% after Deductible
40% after Deductible
20% after Deductible
40% after Deductible
30% after Deductible
Hospital and Surgery (inpatient including maternity)
20% after Deductible
40% after Deductible
20% after Deductible
40% after Deductible
30% after Deductible
Hospital and Surgery (Outpatient including maternity)
20% after Deductible
40% after Deductible
20% after Deductible
40% after Deductible
30% after Deductible
Emergency Room
516
514
20% after in-Network Deductible Plus $150 Copay ((copay waived if admitted))
20% after in-Network Deductible Plus $150 Copay ((copay waived if admitted))
20% after in-Network Deductible Plus $150 Copay ((copay waived if admitted))
20% after in-Network Deductible Plus $150 Copay ((copay waived if admitted))
30% after in-Network Deductible Plus $150 Copay ((copay waived if admitted))
518
410
High Deductible PPO
High Deductible PPO With HSA
HMO
Low Deductible PPO
522
Coinsurance
Copay
528
Emergency Room
526
528
$150
$150
Lifetime Max
Medium Deductible PPO
OOA
PPO
Select a value to see details
420

FIG. 5

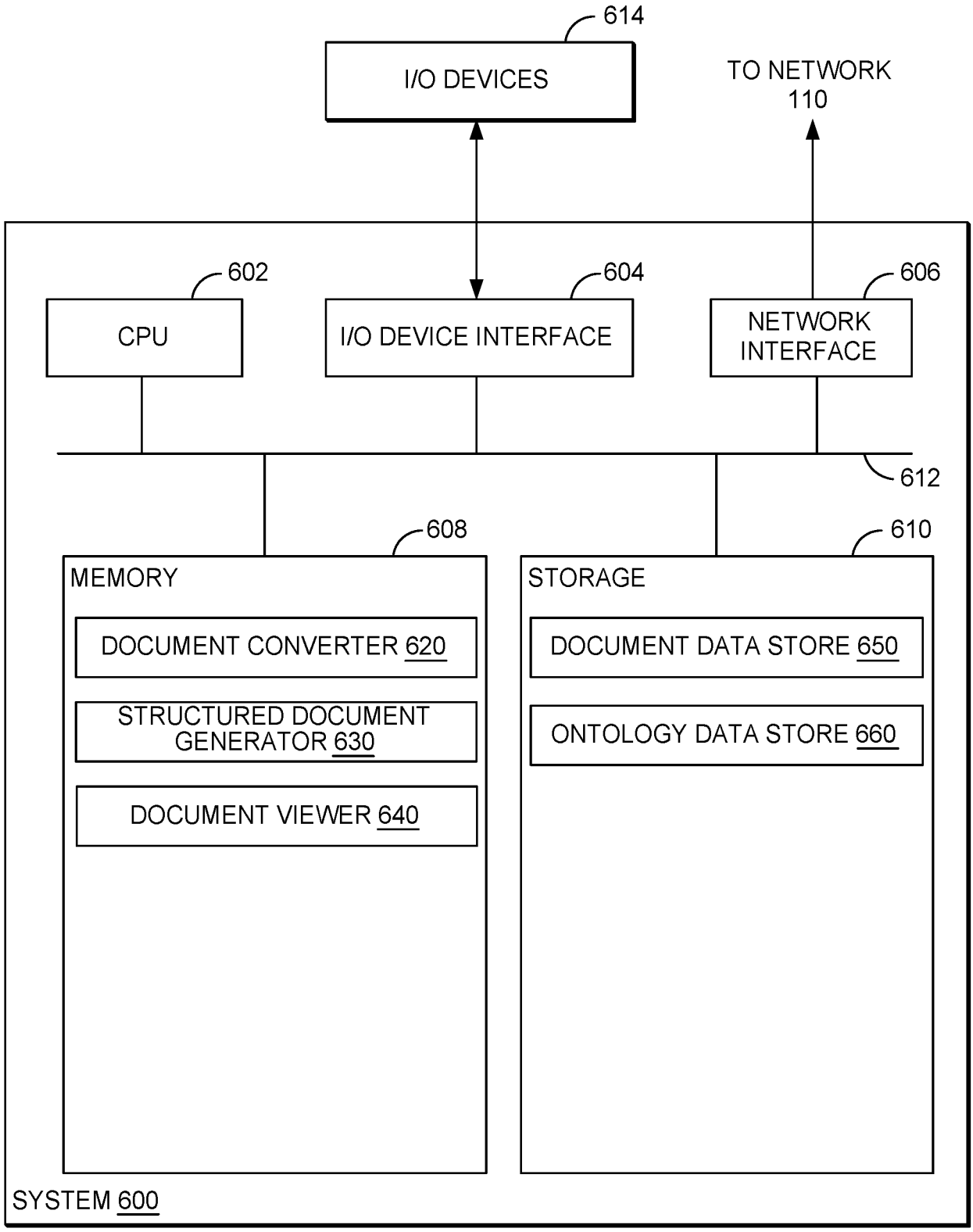

614
I/O DEVICES
TO NETWORK
110
602
CPU
604
I/O DEVICE INTERFACE
606
NETWORK
INTERFACE
612
608
MEMORY
DOCUMENT CONVERTER 620
STRUCTURED DOCUMENT
GENERATOR 630
DOCUMENT VIEWER 640
610
STORAGE
DOCUMENT DATA STORE 650
ONTOLOGY DATA STORE 660
SYSTEM 600

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB2021/050116

A. CLASSIFICATION OF SUBJECT MATTER

G06F 16/36(2019.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F; G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT,WPI,EPODOC,IEEE,CNKI,GOOGLE: unstructure+, structure+, extract+, NLP, ontology, position, location, link+, model+, convert+, text

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 110443571 A (ALIBABA GROUP HOLDING LTD.) 12 November 2019 (2019-11-12) description, paragraphs [0035]-[0065], figures 2-3	1-3,9-11,17-18
A	US 2016247087 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 25 August 2016 (2016-08-25) the whole document	1-20
A	US 2003177112 A1 (GARDNER, Steve) 18 September 2003 (2003-09-18) the whole document	1-20
A	US 10482384 B1 (BABYLON PARTNERS LIMITED) 19 November 2019 (2019-11-19) the whole document	1-20

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

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

Date of the actual completion of the international search	Date of mailing of the international search report
07 April 2021	**16 April 2021**
Name and mailing address of the ISA/CN **National Intellectual Property Administration, PRC** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088** **China** Facsimile No. **(86-10)62019451**	Authorized officer **GAO,Dandan** Telephone No. **86-(10)-53961301**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IB2021/050116

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	110443571	A	12 November 2019	None			
US	2016247087	A1	25 August 2016	CN	105912559	A	31 August 2016
US	2003177112	A1	18 September 2003	AU	2003230091	A1	08 October 2003
				WO	03081468	A2	02 October 2003
US	10482384	B1	19 November 2019	EP	3654227	A1	20 May 2020
				CN	111199795	A	26 May 2020