



US 20180121603A1

(19) **United States**

(12) **Patent Application Publication**

Devarakonda et al.

(10) **Pub. No.: US 2018/0121603 A1**

(43) **Pub. Date: May 3, 2018**

(54) **IDENTIFICATION OF RELATED ELECTRONIC MEDICAL RECORD DOCUMENTS IN A QUESTION AND ANSWER SYSTEM**

(52) **U.S. Cl.**
CPC **G06F 19/322** (2013.01); **G06F 17/30654** (2013.01); **G06F 17/30401** (2013.01); **G06F 19/345** (2013.01)

(71) Applicant: **International Business Machines Corporation**, Armonk, NY (US)

(57) **ABSTRACT**

(72) Inventors: **Murthy V. Devarakonda**, Peekskill, NY (US); **Jennifer J. Liang**, New York, NY (US); **Siddharth A. Patwardhan**, Yorktown Heights, NY (US); **Preethi Raghavan**, Cambridge, MA (US)

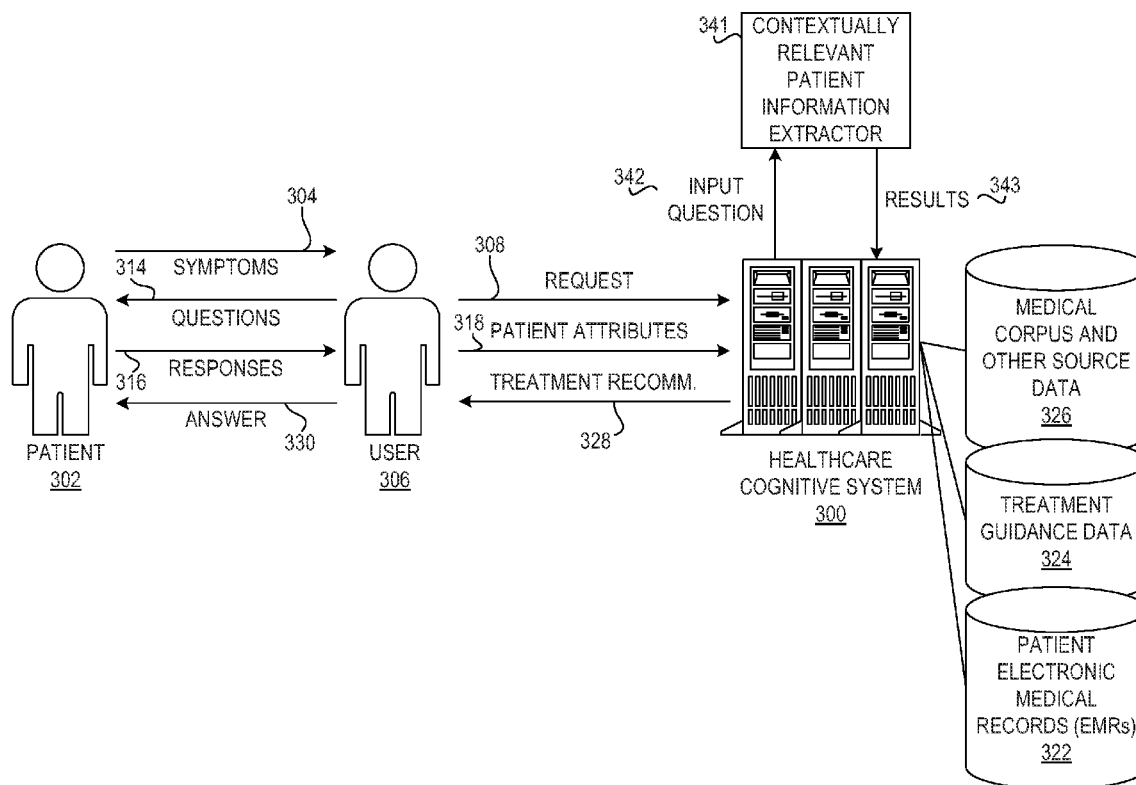
(21) Appl. No.: **15/336,261**

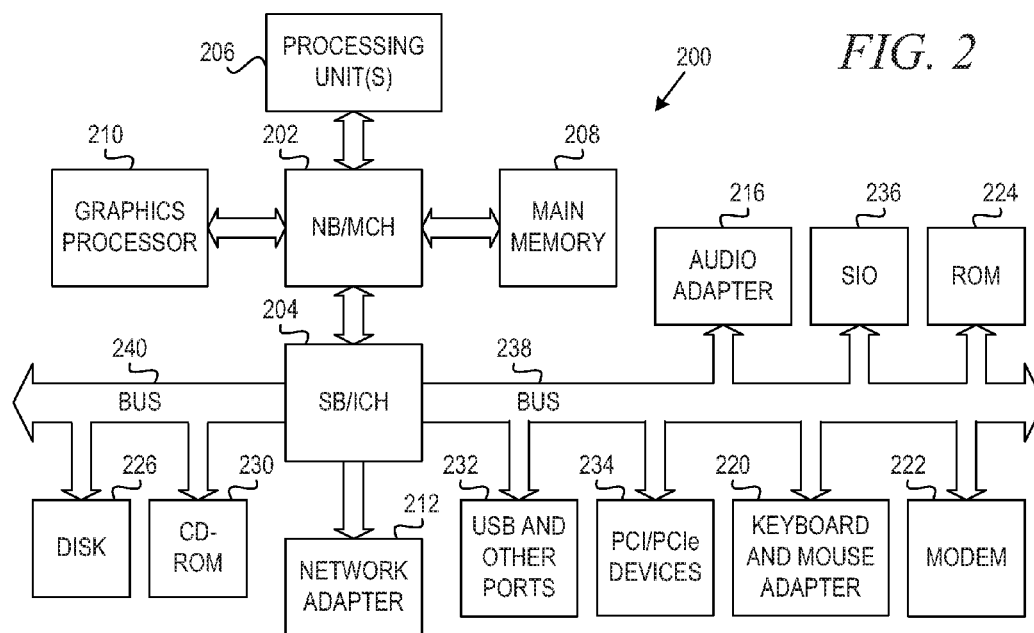
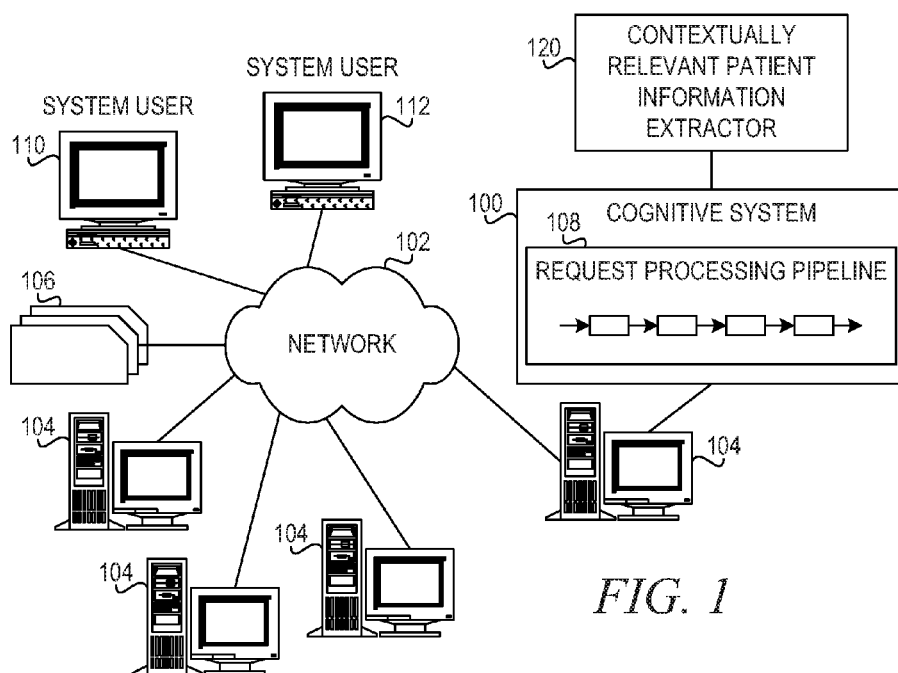
(22) Filed: **Oct. 27, 2016**

Publication Classification

(51) **Int. Cl.**
G06F 19/00 (2006.01)
G06F 17/30 (2006.01)

A contextually relevant patient information extractor is provided that receives an input question directed to medical information about a patient; analyzes a patient's electronic medical records (EMRs) to identify an initial entry in the patient's EMRs corresponding to a candidate answer to the input question; analyzes a context of the patient's EMRs based on the initial entry to identify entries in the patient's EMR that are contextually connected to the initial entry; performs question answering analysis on the initial entry and entries that are contextually connected to the initial entry to identify one or more candidate answers to the input question; and outputs a final answer to the input question based on the question answering analysis.





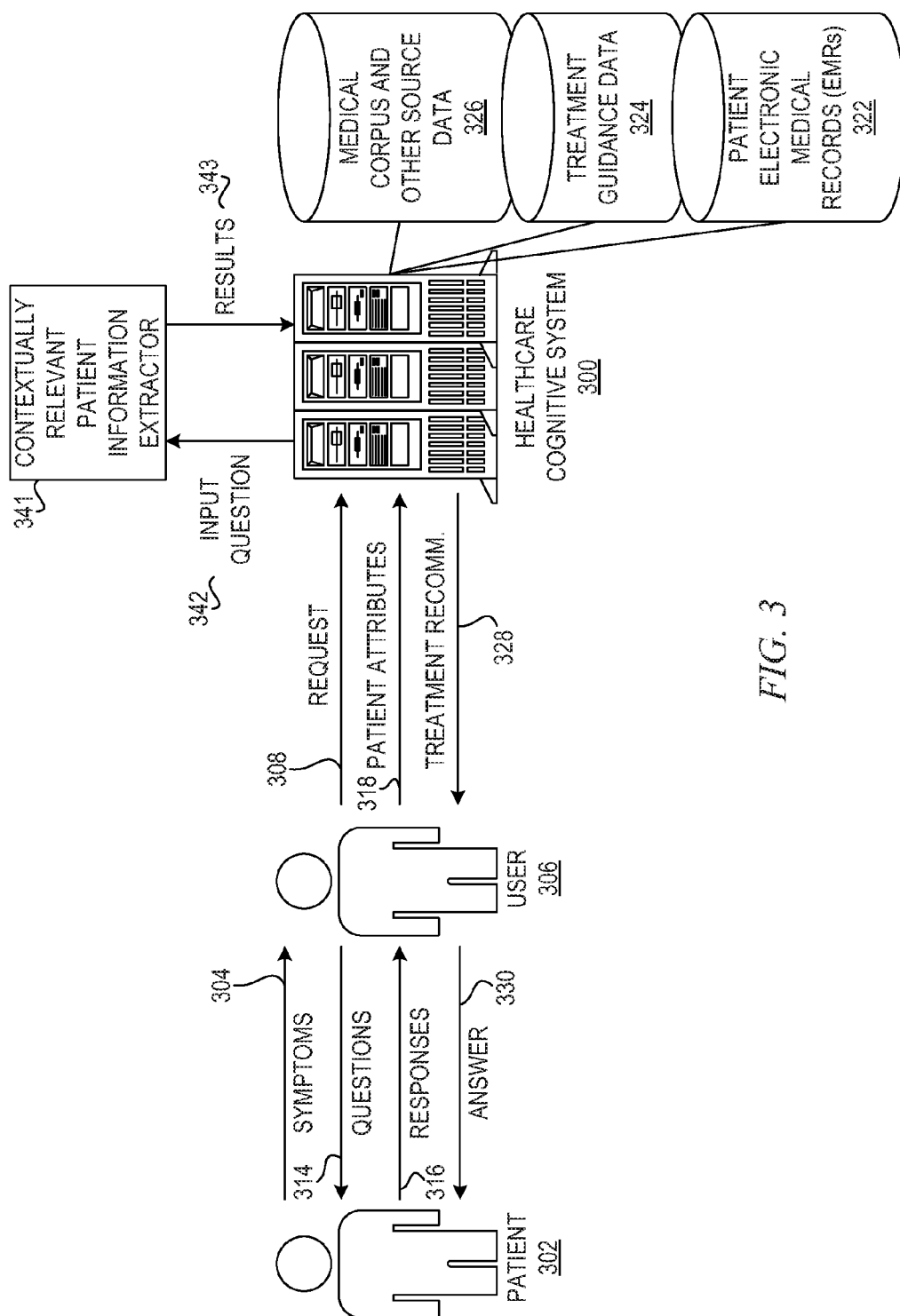


FIG. 3

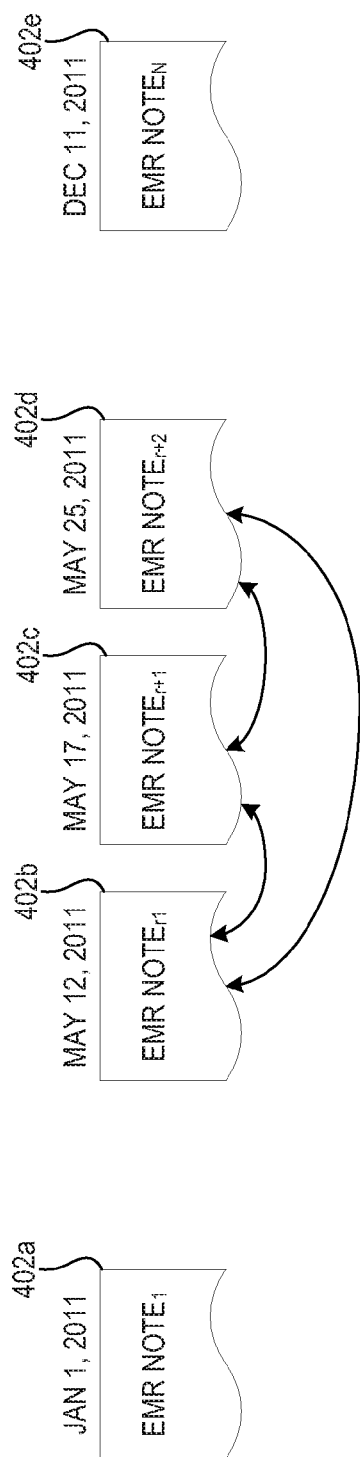


FIG. 4A

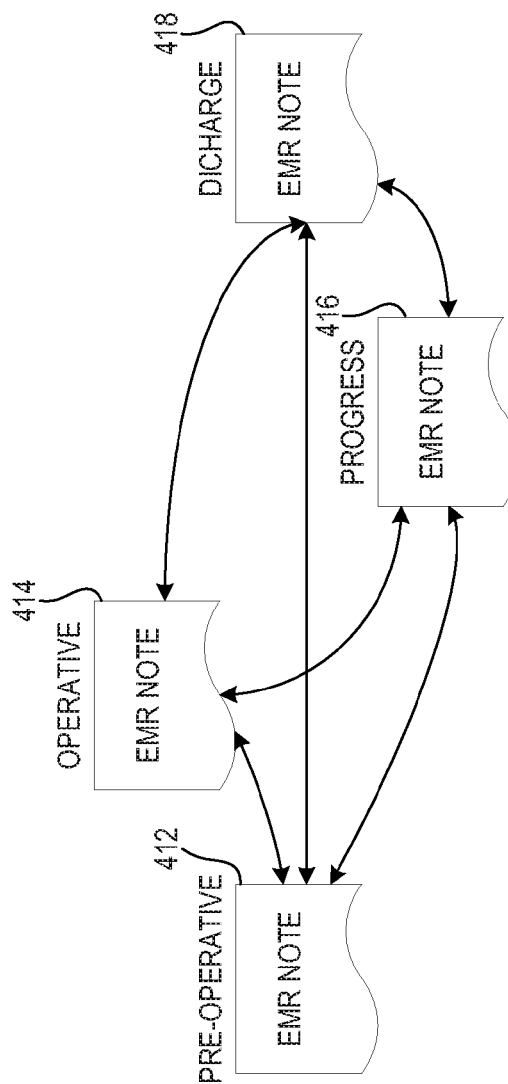


FIG. 4B

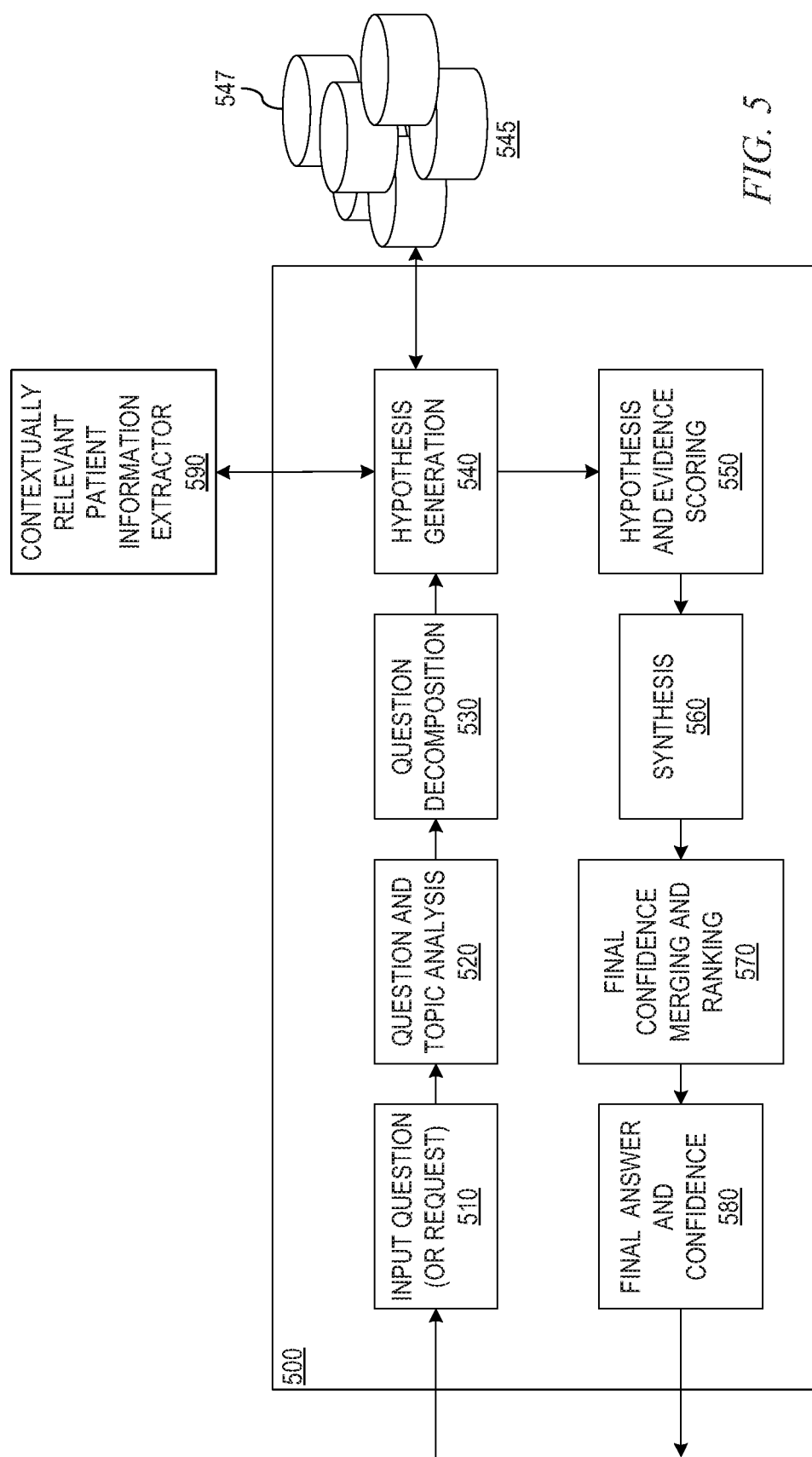
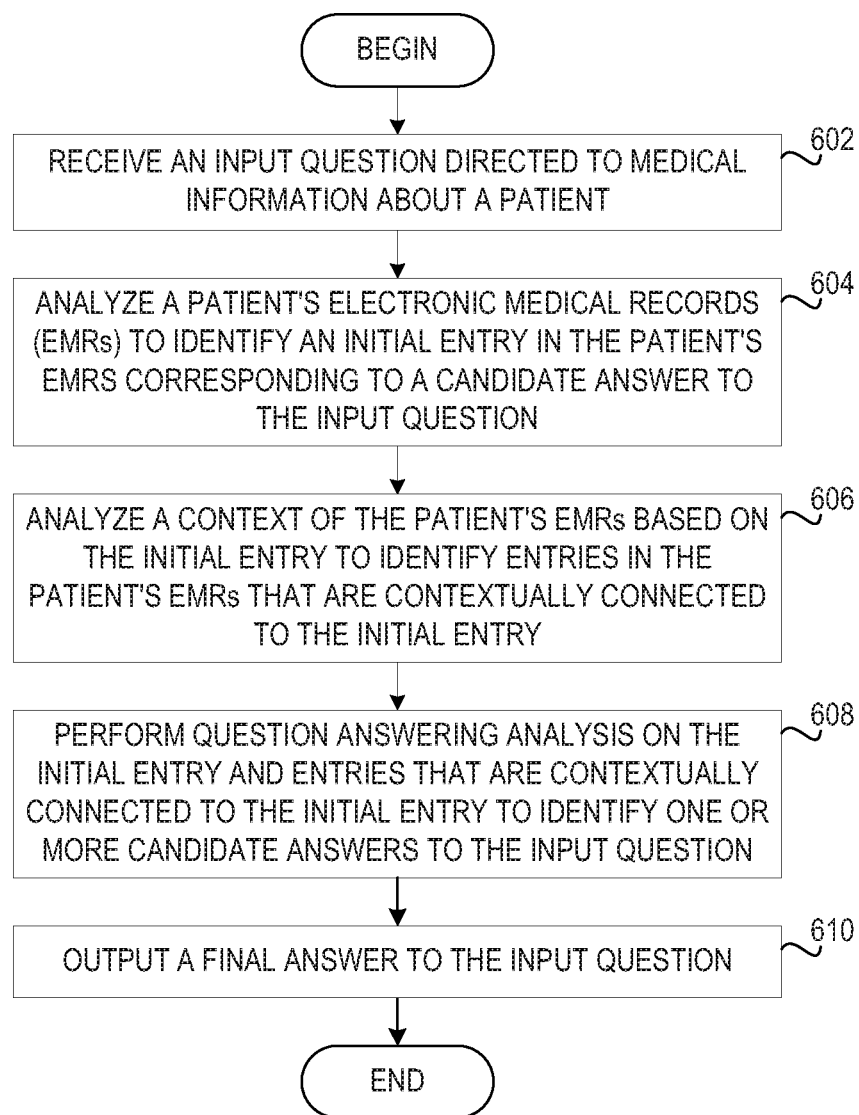


FIG. 5

*FIG. 6*

IDENTIFICATION OF RELATED ELECTRONIC MEDICAL RECORD DOCUMENTS IN A QUESTION AND ANSWER SYSTEM

BACKGROUND

[0001] The present application relates generally to an improved data processing apparatus and method and more specifically to mechanisms for identifying related medical record documents in a Question Answering (QA) system.

[0002] Decision-support systems exist in many different industries where human experts require assistance in retrieving and analyzing information. An example that will be used throughout this application is a diagnosis system employed in the healthcare industry. Diagnosis systems can be classified into systems that use structured knowledge, systems that use unstructured knowledge, and systems that use clinical decision formulas, rules, trees, or algorithms. The earliest diagnosis systems used structured knowledge or classical, manually constructed knowledge bases. The Internist-I system developed in the 1970s uses disease-finding relations and disease-disease relations. The MYCIN system for diagnosing infectious diseases, also developed in the 1970s, uses structured knowledge in the form of production rules, stating that if certain facts are true, then one can conclude certain other facts with a given certainty factor. DXplain, developed starting in the 1980s, uses structured knowledge similar to that of Internist-I, but adds a hierarchical lexicon of findings.

[0003] Iliad, developed starting in the 1990s, adds more sophisticated probabilistic reasoning where each disease has an associated a priori probability of the disease (in the population for which Iliad was designed), and a list of findings along with the fraction of patients with the disease who have the finding (sensitivity), and the fraction of patients without the disease who have the finding (1-specificity).

[0004] In 2000, diagnosis systems using unstructured knowledge started to appear. These systems use some structuring of knowledge such as, for example, entities such as findings and disorders being tagged in documents to facilitate retrieval. ISABEL, for example, uses Autonomy information retrieval software and a database of medical textbooks to retrieve appropriate diagnoses given input findings. Autonomy Aumience uses the Autonomy technology to retrieve diagnoses given findings and organizes the diagnoses by body system. First CONSULT allows one to search a large collection of medical books, journals, and guidelines by chief complaints and age group to arrive at possible diagnoses. PEPID DDX is a diagnosis generator based on PEPID's independent clinical content.

[0005] Clinical decision rules have been developed for a number of medical disorders, and computer systems have been developed to help practitioners and patients apply these rules. The Acute Cardiac Ischemia Time-Insensitive Predictive Instrument (ACI-TIPI) takes clinical and Electrocardiography (ECG) features as input and produces probability of acute cardiac ischemia as output to assist with triage of patients with chest pain or other symptoms suggestive of acute cardiac ischemia. ACI-TIPI is incorporated into many commercial heart monitors/defibrillators. The CaseWalker system uses a four-item questionnaire to diagnose major depressive disorder. The PKC Advisor provides guidance on 98 patient problems such as abdominal pain and vomiting.

SUMMARY

[0006] This Summary is provided to introduce a selection of concepts in a simplified form that are further described herein in the Detailed Description. This Summary is not intended to identify key factors or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

[0007] In one illustrative embodiment, a method is provided, in a data processing system that comprises at least one processor and at least one memory, the at least one memory comprising instructions executed by the at least one processor to cause the at least one processor to implement a contextually relevant patient information extractor. The illustrative embodiment receives an input question directed to medical information about a patient. The illustrative embodiment analyzes a patient's electronic medical records (EMRs) to identify an initial entry in the patient's EMRs corresponding to a candidate answer to the input question. The illustrative embodiment analyzes a context of the patient's EMRs based on the initial entry to identify entries in the patient's EMR that are contextually connected to the initial entry. The illustrative embodiment performs question answering analysis on the initial entry and entries that are contextually connected to the initial entry to identify one or more candidate answers to the input question. The illustrative embodiment outputs a final answer to the input question based on the question answering analysis.

[0008] In other illustrative embodiments, a computer program product comprising a computer useable or readable medium having a computer readable program is provided. The computer readable program, when executed on a computing device, causes the computing device to perform various ones of, and combinations of, the operations outlined above with regard to the method illustrative embodiment.

[0009] In yet another illustrative embodiment, a system/apparatus is provided. The system/apparatus may comprise one or more processors and a memory coupled to the one or more processors. The memory may comprise instructions which, when executed by the one or more processors, cause the one or more processors to perform various ones of, and combinations of, the operations outlined above with regard to the method illustrative embodiment.

[0010] These and other features and advantages of the present invention will be described in, or will become apparent to those of ordinary skill in the art in view of, the following detailed description of the example embodiments of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The invention, as well as a preferred mode of use and further objectives and advantages thereof, will best be understood by reference to the following detailed description of illustrative embodiments when read in conjunction with the accompanying drawings, wherein:

[0012] FIG. 1 depicts a schematic diagram of one illustrative embodiment of a cognitive healthcare system in a computer network;

[0013] FIG. 2 is a block diagram of an example data processing system in which aspects of the illustrative embodiments are implemented;

[0014] FIG. 3 is an example diagram illustrating an interaction of elements of a healthcare cognitive system in accordance with one illustrative embodiment;

[0015] FIGS. 4A-4B depict exemplary illustrations of the process performed by a contextually relevant patient information extractor in linking two or more documents together based on relevancy in accordance with an illustrative embodiment;

[0016] FIG. 5 illustrates a cognitive healthcare system implementing a Question and Answer (QA) or request processing pipeline for processing an input question or request in accordance with one illustrative embodiment; and

[0017] FIG. 6 is a flowchart illustrating operation of a cognitive relevant patient information extractor in accordance with an illustrative embodiment.

DETAILED DESCRIPTION

[0018] The strengths of current cognitive systems, such as current medical diagnosis, patient health management, patient treatment recommendation systems, law enforcement investigation systems, and other decision support systems, are that they can provide insights that improve the decision making performed by human beings. For example, in the medical context, such cognitive systems may improve medical practitioners' diagnostic hypotheses, can help medical practitioners avoid missing important diagnoses, and can assist medical practitioners with determining appropriate treatments for specific diseases. However, current systems still suffer from significant drawbacks, which should be addressed, in order to make such systems more accurate and usable for a variety of applications as well as more representative of the way in which human beings make decisions, such as diagnosing and treating patients. In particular, one drawback of current systems is the accuracy of question answering based on a patient's electronic medical records (EMRs).

[0019] The illustrative embodiments provide mechanisms for improving question answering system accuracy over EMRs by exploiting inter-document relationships. Question Answering (QA) systems rely on evidence from a collection of text documents to generate and score candidate answers to natural language questions. For question answering over EMRs, these documents are clinical notes pertaining to a particular patient. Because the documents pertain to the same patient, documents within this collection are related to one another in various different ways. The illustrative embodiments improve the accuracy and coverage of the QA system by exploiting "inter-document" relationships across the EMR documents related to the patient. In some illustrative embodiments, the illustrative embodiments address the problem by narrowing of the scope of portions of the patient's EMRs and/or medical documentation to determine a condition or reasons for a medical condition of a patient. That is, the illustrative embodiments are directed to focusing analysis on the most relevant portions of a patient's EMRs and medical documentation associated with the relevant portions of the patient's EMR. The illustrative embodiments augment existing QA technology with a mechanism for identifying portions of the patient's EMRs that are most relevant to a particular question for answering an input question. The most relevant portions are identified through inter-document relations encoded in a context graph, which is used in searching and scoring relationships of medical factors as potential answers to the input question.

[0020] Before beginning the discussion of the various aspects of the illustrative embodiments in more detail, it should first be appreciated that throughout this description the term "mechanism" will be used to refer to elements of the present invention that perform various operations, functions, and the like. A "mechanism," as the term is used herein, may be an implementation of the functions or aspects of the illustrative embodiments in the form of an apparatus, a procedure, or a computer program product. In the case of a procedure, the procedure is implemented by one or more devices, apparatus, computers, data processing systems, or the like. In the case of a computer program product, the logic represented by computer code or instructions embodied in or on the computer program product is executed by one or more hardware devices in order to implement the functionality or perform the operations associated with the specific "mechanism." Thus, the mechanisms described herein may be implemented as specialized hardware, software executing on general-purpose hardware, software instructions stored on a medium such that the instructions are readily executable by specialized or general-purpose hardware, a procedure or method for executing the functions, or a combination of any of the above.

[0021] The present description and claims may make use of the terms "a," "at least one of," and "one or more of" with regard to particular features and elements of the illustrative embodiments. It should be appreciated that these terms and phrases are intended to state that there is at least one of the particular feature or element present in the particular illustrative embodiment, but that more than one can also be present. That is, these terms/phrases are not intended to limit the description or claims to a single feature/element being present or require that a plurality of such features/elements be present. To the contrary, these terms/phrases only require at least a single feature/element with the possibility of a plurality of such features/elements being within the scope of the description and claims.

[0022] Moreover, it should be appreciated that the use of the term "engine," if used herein with regard to describing embodiments and features of the invention, is not intended to be limiting of any particular implementation for accomplishing and/or performing the actions, steps, processes, etc., attributable to and/or performed by the engine. An engine may be, but is not limited to, software, hardware and/or firmware or any combination thereof that performs the specified functions including, but not limited to, any use of a general and/or specialized processor in combination with appropriate software loaded or stored in a machine readable memory and executed by the processor. Further, any name associated with a particular engine is, unless otherwise specified, for purposes of convenience of reference and not intended to be limiting to a specific implementation. Additionally, any functionality attributed to an engine may be equally performed by multiple engines, incorporated into and/or combined with the functionality of another engine of the same or different type, or distributed across one or more engines of various configurations.

[0023] In addition, it should be appreciated that the following description uses a plurality of various examples for various elements of the illustrative embodiments to further illustrate example implementations of the illustrative embodiments and to aid in the understanding of the mechanisms of the illustrative embodiments. These examples intended to be non-limiting and are not exhaustive of the

various possibilities for implementing the mechanisms of the illustrative embodiments. It will be apparent to those of ordinary skill in the art in view of the present description that there are many other alternative implementations for these various elements that may be utilized in addition to, or in replacement of, the examples provided herein without departing from the spirit and scope of the present invention.

[0024] As noted above, the present invention provides mechanisms for improving question answering system accuracy over EMRs by exploiting inter-document relationships across the EMR documents related to the patient. The mechanisms narrow the scope of portions of the patient's EMRs and/or medical documentation to determine a condition or reasons for a medical condition of a patient. That is, the mechanisms focus analysis on the most relevant portions of a patient's EMRs and medical documentation associated with the relevant portions of the patient's EMR. Thus, the mechanisms identify portions of the patient's EMRs that are most relevant to a particular question for answering an input question. The mechanisms utilize inter-document relations encoded in a context graph to identify the most relevant portions, which are then used in searching and scoring relationships of medical factors as potential answers to the input question.

[0025] The illustrative embodiments may be utilized in many different types of data processing environments. In order to provide a context for the description of the specific elements and functionality of the illustrative embodiments, FIGS. 1-5 are provided hereafter as example environments in which aspects of the illustrative embodiments may be implemented. It should be appreciated that FIGS. 1-5 are only examples and are not intended to assert or imply any limitation with regard to the environments in which aspects or embodiments of the present invention may be implemented. Many modifications to the depicted environments may be made without departing from the spirit and scope of the present invention.

[0026] FIGS. 1-5 are directed to describing an example cognitive system for healthcare applications (also referred to herein as a "healthcare cognitive system") which implements a request processing pipeline, such as a Question Answering (QA) pipeline (also referred to as a Question/Answer pipeline or Question and Answer pipeline) for example, request processing methodology, and request processing computer program product with which the mechanisms of the illustrative embodiments are implemented. These requests may be provided as structured or unstructured request messages, natural language questions, or any other suitable format for requesting an operation to be performed by the healthcare cognitive system. As described in more detail hereafter, the particular healthcare application that is implemented in the cognitive system of the present invention is a healthcare application for providing medical treatment recommendations for patients based on their specific features as obtained from various sources, e.g., patient electronic medical records (EMRs), patient questionnaires, etc. In particular, the mechanisms of the present invention provide a mechanism for improving question answering system accuracy over EMRs by exploiting inter-document relationships.

[0027] It should be appreciated that the healthcare cognitive system, while shown as having a single request processing pipeline in the examples hereafter, may in fact have multiple request processing pipelines. Each request process-

ing pipeline may be separately trained and/or configured to process requests associated with different domains or be configured to perform the same or different analysis on input requests (or questions in implementations using a QA pipeline), depending on the desired implementation. For example, in some cases, a first request processing pipeline may be trained to operate on input requests directed to a first medical malady domain (e.g., various types of blood diseases) while another request processing pipeline may be trained to answer input requests in another medical malady domain (e.g., various types of cancers). In other cases, for example, the request processing pipelines may be configured to provide different types of cognitive functions or support different types of healthcare applications, such as one request processing pipeline being used for patient diagnosis, another request processing pipeline being configured for medical treatment recommendation, another request processing pipeline being configured for patient monitoring, etc.

[0028] Moreover, each request processing pipeline may have their own associated corpus or corpora that they ingest and operate on, e.g., one corpus for blood disease domain documents and another corpus for cancer diagnostics domain related documents in the above examples. In some cases, the request processing pipelines may each operate on the same domain of input questions but may have different configurations, e.g., different annotators or differently trained annotators, such that different analysis and potential answers are generated. The healthcare cognitive system may provide additional logic for routing input questions to the appropriate request processing pipeline, such as based on a determined domain of the input request, combining and evaluating final results generated by the processing performed by multiple request processing pipelines, and other control and interaction logic that facilitates the utilization of multiple request processing pipelines.

[0029] As noted above, one type of request processing pipeline with which the mechanisms of the illustrative embodiments may be utilized is a Question Answering (QA) pipeline. The description of example embodiments of the present invention hereafter will utilize a QA pipeline as an example of a request processing pipeline that may be augmented to include mechanisms in accordance with one or more illustrative embodiments. It should be appreciated that while the present invention will be described in the context of the cognitive system implementing one or more QA pipelines that operate on an input question, the illustrative embodiments are not limited to such. Rather, the mechanisms of the illustrative embodiments may operate on requests that are not posed as "questions" but are formatted as requests for the cognitive system to perform cognitive operations on a specified set of input data using the associated corpus or corpora and the specific configuration information used to configure the cognitive system. For example, rather than asking a natural language question of "What diagnosis applies to patient P?", the cognitive system may instead receive a request of "generate diagnosis for patient P," or the like. It should be appreciated that the mechanisms of the QA system pipeline may operate on requests in a similar manner to that of input natural language questions with minor modifications. In fact, in some cases, a request may be converted to a natural language question for processing by the QA system pipelines if desired for the particular implementation.

[0030] As will be discussed in greater detail hereafter, the illustrative embodiments may be integrated in, augment, and extend the functionality of these QA pipeline, or request processing pipeline, mechanisms of a healthcare cognitive system with regard to providing a medical malady independent treatment recommendation system which may receive an input question regarding the recommended treatment for a specific patient and may utilize the QA pipeline mechanisms to evaluate patient information and other medical information in one or more corpora of medical information to determine the most appropriate treatment for the specific patient.

[0031] Thus, it is important to first have an understanding of how cognitive systems and question and answer creation in a cognitive system implementing a QA pipeline is implemented before describing how the mechanisms of the illustrative embodiments are integrated in and augment such cognitive systems and request processing pipeline, or QA pipeline, mechanisms. It should be appreciated that the mechanisms described in FIGS. 1-5 are only examples and are not intended to state or imply any limitation with regard to the type of cognitive system mechanisms with which the illustrative embodiments are implemented. Many modifications to the example cognitive system shown in FIGS. 1-5 may be implemented in various embodiments of the present invention without departing from the spirit and scope of the present invention.

[0032] As an overview, a cognitive system is a specialized computer system, or set of computer systems, configured with hardware and/or software logic (in combination with hardware logic upon which the software executes) to emulate human cognitive functions. These cognitive systems apply human-like characteristics to conveying and manipulating ideas which, when combined with the inherent strengths of digital computing, can solve problems with high accuracy and resilience on a large scale. A cognitive system performs one or more computer-implemented cognitive operations that approximate a human thought process as well as enable people and machines to interact in a more natural manner to extend and magnify human expertise and cognition. A cognitive system comprises artificial intelligence logic, such as natural language processing (NLP) based logic, for example, and machine learning logic, which may be provided as specialized hardware, software executed on hardware, or any combination of specialized hardware and software executed on hardware. The logic of the cognitive system implements the cognitive operation(s), examples of which include, but are not limited to, question answering, identification of related concepts within different portions of content in a corpus, intelligent search algorithms, such as Internet web page searches, for example, medical diagnostic and treatment recommendations, and other types of recommendation generation, e.g., items of interest, to a particular user, potential new contact recommendations, or the like.

[0033] IBM Watson™ is an example of one such cognitive system that can process human readable language and identify inferences between text passages with human-like high accuracy at speeds far faster than human beings and on a larger scale. In general, such cognitive systems are able to perform the following functions:

[0034] Navigate the complexities of human language and understanding;

[0035] Ingest and process vast amounts of structured and unstructured data;

[0036] Generate and evaluate hypothesis;

[0037] Weigh and evaluate responses that are based only on relevant evidence;

[0038] Provide situation-specific advice, insights, and guidance;

[0039] Improve knowledge and learn with each iteration and interaction. through machine learning processes;

[0040] Enable decision making at the point of impact (contextual guidance);

[0041] Scale in proportion to the task;

[0042] Extend and magnify human expertise and cognition;

[0043] Identify resonating, human-like attributes and traits from natural language;

[0044] Deduce various language specific or agnostic attributes from natural language;

[0045] High degree of relevant recollection from data points (images, text, voice) (memorization and recall);

[0046] Predict and sense with situational awareness that mimic human cognition based on experiences; and/or

[0047] Answer questions based on natural language and specific evidence.

[0048] In one aspect, cognitive systems provide mechanisms for answering questions posed to these cognitive systems using a Question Answering pipeline or system (QA system) and/or process requests that may or may not be posed as natural language questions. The QA pipeline or system is an artificial intelligence application executing on data processing hardware that answers questions pertaining to a given subject matter domain presented in natural language. The QA pipeline receives inputs from various sources including input over a network, a corpus of electronic documents or other data, data from a content creator, information from one or more content users, and other such inputs from other possible sources of input. Data storage devices store the corpus of data. A content creator creates content in a document for use as part of a corpus of data with the QA pipeline. The document may include any file, text, article, or source of data for use in the QA system. For example, a QA pipeline accesses a body of knowledge about the domain, or subject matter area, e.g., financial domain, medical domain, legal domain, etc., where the body of knowledge (knowledgebase) can be organized in a variety of configurations, e.g., a structured repository of domain-specific information, such as ontologies, or unstructured data related to the domain, or a collection of natural language documents about the domain.

[0049] Content users input questions to a cognitive system, which implements the QA pipeline. The QA pipeline then answers the input questions using the content in the corpus of data by evaluating documents, sections of documents, portions of data in the corpus, or the like. When a process evaluates a given section of a document for semantic content, the process can use a variety of conventions to query such document from the QA pipeline, e.g., sending the query to the QA pipeline as a well-formed question which is then interpreted by the QA pipeline and a response is provided containing one or more answers to the question. Semantic content is content based on the relation between signifiers, such as words, phrases, signs, and symbols, and what they stand for, their denotation, or connotation. In other

words, semantic content is content that interprets an expression, such as by using Natural Language Processing.

[0050] As will be described in greater detail hereafter, the QA pipeline receives an input question, parses the question to extract the major features of the question, uses the extracted features to formulate queries, and then applies those queries to the corpus of data. Based on the application of the queries to the corpus of data, the QA pipeline generates a set of hypotheses, or candidate answers to the input question, by looking across the corpus of data for portions of the corpus of data that have some potential for containing a valuable response to the input question. The QA pipeline then performs deep analysis on the language of the input question and the language used in each of the portions of the corpus of data found during the application of the queries using a variety of reasoning algorithms. There may be hundreds or even thousands of reasoning algorithms applied, each of which performs different analysis, e.g., comparisons, natural language analysis, lexical analysis, or the like, and generates a score. For example, some reasoning algorithms may look at the matching of terms and synonyms within the language of the input question and the found portions of the corpus of data. Other reasoning algorithms may look at temporal or spatial features in the language, while others may evaluate the source of the portion of the corpus of data and evaluate its veracity.

[0051] The scores obtained from the various reasoning algorithms indicate the extent to which the potential response is inferred by the input question based on the specific area of focus of that reasoning algorithm. Each resulting score is then weighted against a statistical model. The statistical model captures how well the reasoning algorithm performed at establishing the inference between two similar passages for a particular domain during the training period of the QA pipeline. The statistical model is used to summarize a level of confidence that the QA pipeline has regarding the evidence that the potential response, i.e. candidate answer, is inferred by the question. This process is repeated for each of the candidate answers until the QA pipeline identifies candidate answers that surface as being significantly stronger than others and thus, generates a final answer, or ranked set of answers, for the input question.

[0052] As mentioned above, QA pipeline mechanisms operate by accessing information from a corpus of data or information (also referred to as a corpus of content), analyzing it, and then generating answer results based on the analysis of this data. Accessing information from a corpus of data typically includes a database query that answers questions about what is in a collection of structured records, and a search that delivers a collection of document links in response to a query against a collection of unstructured data (text, markup language, etc.). Conventional question answering systems are capable of generating answers based on the corpus of data and the input question, verifying answers to a collection of questions for the corpus of data, correcting errors in digital text using a corpus of data, and selecting answers to questions from a pool of potential answers, i.e. candidate answers.

[0053] Content creators, such as article authors, electronic document creators, web page authors, document database creators, and the like, determine use cases for products, solutions, and services described in such content before writing their content. Consequently, the content creators know what questions the content is intended to answer in a

particular topic addressed by the content. Categorizing the questions, such as in terms of roles, type of information, tasks, or the like, associated with the question, in each document of a corpus of data allows the QA pipeline to more quickly and efficiently identify documents containing content related to a specific query. The content may also answer other questions that the content creator did not contemplate that may be useful to content users. The questions and answers may be verified by the content creator to be contained in the content for a given document. These capabilities contribute to improved accuracy, system performance, machine learning, and confidence of the QA pipeline. Content creators, automated tools, or the like, annotate or otherwise generate metadata for providing information useable by the QA pipeline to identify these question and answer attributes of the content.

[0054] Operating on such content, the QA pipeline generates answers for input questions using a plurality of intensive analysis mechanisms that evaluate the content to identify the most probable answers, i.e. candidate answers, for the input question. The most probable answers are output as a ranked listing of candidate answers ranked according to their relative scores or confidence measures calculated during evaluation of the candidate answers, as a single final answer having a highest ranking score or confidence measure, or which is a best match to the input question, or a combination of ranked listing and final answer.

[0055] FIG. 1 depicts a schematic diagram of one illustrative embodiment of a cognitive system **100** implementing a request processing pipeline **108**, which in some embodiments may be a question answering (QA) pipeline, in a computer network **102**. For purposes of the present description, it will be assumed that the request processing pipeline **108** is implemented as a QA pipeline that operates on structured and/or unstructured requests in the form of input questions. One example of a question processing operation which may be used in conjunction with the principles described herein is described in U.S. Patent Application Publication No. 2011/0125734, which is herein incorporated by reference in its entirety. The cognitive system **100** is implemented on one or more computing devices **104** (comprising one or more processors and one or more memories, and potentially any other computing device elements generally known in the art including buses, storage devices, communication interfaces, and the like) connected to the computer network **102**. The network **102** includes multiple computing devices **104** in communication with each other and with other devices or components via one or more wired and/or wireless data communication links, where each communication link comprises one or more of wires, routers, switches, transmitters, receivers, or the like. The cognitive system **100** and network **102** enables question processing and answer generation (QA) functionality for one or more cognitive system users via their respective computing devices **110-112**. Other embodiments of the cognitive system **100** may be used with components, systems, sub-systems, and/or devices other than those that are depicted herein.

[0056] The cognitive system **100** is configured to implement a QA pipeline **108** that receive inputs from various sources. For example, the cognitive system **100** receives input from the network **102**, a corpus of electronic documents **106**, cognitive system users, and/or other data and other possible sources of input. In one embodiment, some or

all of the inputs to the cognitive system **100** are routed through the network **102**. The various computing devices **104** on the network **102** include access points for content creators and QA system users. Some of the computing devices **104** include devices for a database storing the corpus of data **106** (which is shown as a separate entity in FIG. 1 for illustrative purposes only). Portions of the corpus of data **106** may also be provided on one or more other network attached storage devices, in one or more databases, or other computing devices not explicitly shown in FIG. 1. The network **102** includes local network connections and remote connections in various embodiments, such that the cognitive system **100** may operate in environments of any size, including local and global, e.g., the Internet.

[0057] In one embodiment, the content creator creates content in a document of the corpus of data **106** for use as part of a corpus of data with the cognitive system **100**. The document includes any file, text, article, or source of data for use in the cognitive system **100**. QA system users access the cognitive system **100** via a network connection or an Internet connection to the network **102**, and input questions to the cognitive system **100** that are answered by the content in the corpus of data **106**. In one embodiment, the questions are formed using natural language. The cognitive system **100** parses and interprets the question via a QA pipeline **108**, and provides a response to the cognitive system user, e.g., cognitive system user **110**, containing one or more answers to the question. In some embodiments, the cognitive system **100** provides a response to users in a ranked list of candidate answers while in other illustrative embodiments, the cognitive system **100** provides a single final answer or a combination of a final answer and ranked listing of other candidate answers.

[0058] The cognitive system **100** implements the QA pipeline **108**, which comprises a plurality of stages for processing an input question and the corpus of data **106**. The QA pipeline **108** generates answers for the input question based on the processing of the input question and the corpus of data **106**. The QA pipeline **108** will be described in greater detail hereafter with regard to FIG. 3.

[0059] In some illustrative embodiments, the cognitive system **100** may be the IBM Watson™ cognitive system available from International Business Machines Corporation of Armonk, N.Y., which is augmented with the mechanisms of the illustrative embodiments described hereafter. As outlined previously, a QA pipeline of the IBM Watson™ cognitive system receives an input question, which it then parses to extract the major features of the question, which in turn are then used to formulate queries that are applied to the corpus of data. Based on the application of the queries to the corpus of data, a set of hypotheses, or candidate answers to the input question, are generated by looking across the corpus of data for portions of the corpus of data that have some potential for containing a valuable response to the input question. The QA pipeline of the IBM Watson™ cognitive system then performs deep analysis on the language of the input question and the language used in each of the portions of the corpus of data found during the application of the queries using a variety of reasoning algorithms.

[0060] The scores obtained from the various reasoning algorithms are then weighted against a statistical model that summarizes a level of confidence that the QA pipeline of the IBM Watson™ cognitive system has regarding the evidence that the potential response, i.e. candidate answer, is inferred

by the question. This process is repeated for each of the candidate answers to generate ranked listing of candidate answers, which may then be presented to the user that submitted the input question, or from which a final answer is selected and presented to the user. More information about the QA pipeline of the IBM Watson™ cognitive system may be obtained, for example, from the IBM Corporation website, IBM Redbooks, and the like. For example, information about the QA pipeline of the IBM Watson™ cognitive system can be found in Yuan et al., "Watson and Healthcare," IBM developerWorks, 2011 and "The Era of Cognitive Systems: An Inside Look at IBM Watson and How it Works" by Rob High, IBM Redbooks, 2012.

[0061] As noted above, while the input to the cognitive system **100** from a client device may be posed in the form of a natural language question, the illustrative embodiments are not limited to such. Rather, the input question may in fact be formatted or structured as any suitable type of request which may be parsed and analyzed using structured and/or unstructured input analysis, including but not limited to the natural language parsing and analysis mechanisms of a cognitive system such as IBM Watson™, to determine the basis upon which to perform cognitive analysis and providing a result of the cognitive analysis. In the case of a healthcare based cognitive system, this analysis may involve processing patient medical records, medical guidance documentation from one or more corpora, and the like, to provide a healthcare oriented cognitive system result.

[0062] In the context of the present invention, cognitive system **100** may provide a cognitive functionality for assisting with healthcare based operations. For example, depending upon the particular implementation, the healthcare based operations may comprise patient diagnostics, medical treatment recommendation systems, medical practice management systems, personal patient care plan generation and monitoring, patient electronic medical record (EMR) evaluation for various purposes, such as for identifying patients that are suitable for a medical trial or a particular type of medical treatment, or the like. Thus, the cognitive system **100** may be a healthcare cognitive system **100** that operates in the medical or healthcare type domains and which may process requests for such healthcare operations via the request processing pipeline **108** input as either structured or unstructured requests, natural language input questions, or the like. In one illustrative embodiment, the cognitive system **100** is a medical treatment recommendation system that, for a particular patient, exploits inter-document relationships within the patient's EMRs to improve question answering system accuracy.

[0063] In an illustrative embodiment, the cognitive system **100** implements a contextually relevant patient information extractor **120** for extracting patient information from a patient's EMR. A system user **110**, **112** may provide an input question for contextually relevant patient information to contextually relevant patient information extractor **120**, which runs a formulated query, based on the input question against the patient's EMRs and generates search results. Contextually relevant patient information extractor **120** identifies portions of the patient's EMRs that are most relevant to the query. Contextually relevant patient information extractor **120** encodes latent relationships between documents in the patient's EMRs. Contextually relevant patient information extractor **120** encodes the latent relationships between documents in the patient's EMRs in a

context graph, where each node in the context graph corresponds to a document in the patient's EMRs and edges between pairs of nodes corresponds to encoded latent relationships. Contextually relevant patient information extractor 120 then uses these latent relationships for identifying relevant documents, i.e. documents that have a nearness to each other, such as temporality.

[0064] As noted above, the mechanisms of the illustrative embodiments are rooted in the computer technology arts and are implemented using logic present in such computing or data processing systems. These computing or data processing systems are specifically configured, either through hardware, software, or a combination of hardware and software, to implement the various operations described above. As such, FIG. 2 is provided as an example of one type of data processing system in which aspects of the present invention may be implemented. Many other types of data processing systems may be likewise configured to specifically implement the mechanisms of the illustrative embodiments.

[0065] FIG. 2 is a block diagram of an example data processing system in which aspects of the illustrative embodiments are implemented. Data processing system 200 is an example of a computer, such as server 104 or client 110 in FIG. 1, in which computer usable code or instructions implementing the processes for illustrative embodiments of the present invention are located. In one illustrative embodiment, FIG. 2 represents a server computing device, such as a server 104, which implements a cognitive system 100 and QA system pipeline 108 augmented to include the additional mechanisms of the illustrative embodiments described hereafter.

[0066] In the depicted example, data processing system 200 employs a hub architecture including North Bridge and Memory Controller Hub (NB/MCH) 202 and South Bridge and Input/Output (I/O) Controller Hub (SB/ICH) 204. Processing unit 206, main memory 208, and graphics processor 210 are connected to NB/MCH 202. Graphics processor 210 is connected to NB/MCH 202 through an accelerated graphics port (AGP).

[0067] In the depicted example, local area network (LAN) adapter 212 connects to SB/ICH 204. Audio adapter 216, keyboard and mouse adapter 220, modem 222, read only memory (ROM) 224, hard disk drive (HDD) 226, CD-ROM drive 230, universal serial bus (USB) ports and other communication ports 232, and PCI/PCIe devices 234 connect to SB/ICH 204 through bus 238 and bus 240. PCI/PCIe devices may include, for example, Ethernet adapters, add-in cards, and PC cards for notebook computers. PCI uses a card bus controller, while PCIe does not. ROM 224 may be, for example, a flash basic input/output system (BIOS).

[0068] HDD 226 and CD-ROM drive 230 connect to SB/ICH 204 through bus 240. HDD 226 and CD-ROM drive 230 may use, for example, an integrated drive electronics (IDE) or serial advanced technology attachment (SATA) interface. Super I/O (SIO) device 236 is connected to SB/ICH 204.

[0069] An operating system runs on processing unit 206. The operating system coordinates and provides control of various components within the data processing system 200 in FIG. 2. As a client, the operating system is a commercially available operating system such as Microsoft® Windows 10®. An object-oriented programming system, such as the Java™ programming system, may run in conjunction with

the operating system and provides calls to the operating system from Java™ programs or applications executing on data processing system 200.

[0070] As a server, data processing system 200 may be, for example, an IBM® eServer™ System p® computer system, running the Advanced Interactive Executive (AIX®) operating system or the LINUX® operating system. Data processing system 200 may be a symmetric multiprocessor (SMP) system including a plurality of processors in processing unit 206. Alternatively, a single processor system may be employed.

[0071] Instructions for the operating system, the object-oriented programming system, and applications or programs are located on storage devices, such as HDD 226, and are loaded into main memory 208 for execution by processing unit 206. The processes for illustrative embodiments of the present invention are performed by processing unit 206 using computer usable program code, which is located in a memory such as, for example, main memory 208, ROM 224, or in one or more peripheral devices 226 and 230, for example.

[0072] A bus system, such as bus 238 or bus 240 as shown in FIG. 2, is comprised of one or more buses. Of course, the bus system may be implemented using any type of communication fabric or architecture that provides for a transfer of data between different components or devices attached to the fabric or architecture. A communication unit, such as modem 222 or network adapter 212 of FIG. 2, includes one or more devices used to transmit and receive data. A memory may be, for example, main memory 208, ROM 224, or a cache such as found in NB/MCH 202 in FIG. 2.

[0073] Those of ordinary skill in the art will appreciate that the hardware depicted in FIGS. 1 and 2 may vary depending on the implementation. Other internal hardware or peripheral devices, such as flash memory, equivalent non-volatile memory, or optical disk drives and the like, may be used in addition to or in place of the hardware depicted in FIGS. 1 and 2. Also, the processes of the illustrative embodiments may be applied to a multiprocessor data processing system, other than the SMP system mentioned previously, without departing from the spirit and scope of the present invention.

[0074] Moreover, the data processing system 200 may take the form of any of a number of different data processing systems including client computing devices, server computing devices, a tablet computer, laptop computer, telephone or other communication device, a personal digital assistant (PDA), or the like. In some illustrative examples, data processing system 200 may be a portable computing device that is configured with flash memory to provide non-volatile memory for storing operating system files and/or user-generated data, for example. Essentially, data processing system 200 may be any known or later developed data processing system without architectural limitation.

[0075] FIG. 3 is an example diagram illustrating an interaction of elements of a healthcare cognitive system in accordance with one illustrative embodiment. The example diagram of FIG. 3 depicts an implementation of a healthcare cognitive system 300 that is configured to provide medical treatment recommendations for patients. However, it should be appreciated that this is only an example implementation and other healthcare operations may be implemented in

other embodiments of the healthcare cognitive system 300 without departing from the spirit and scope of the present invention.

[0076] Moreover, it should be appreciated that while FIG. 3 depicts the patient 302 and user 306 as human figures, the interactions with and between these entities may be performed using computing devices, medical equipment, and/or the like, such that entities 302 and 306 may in fact be computing devices, e.g., client computing devices. For example, the interactions 304, 314, 316, and 330 between the patient 302 and the user 306 may be performed orally, e.g., a doctor interviewing a patient, and may involve the use of one or more medical instruments, monitoring devices, or the like, to collect information that may be input to the healthcare cognitive system 300 as patient attributes 318. Interactions between the user 306 and the healthcare cognitive system 300 will be electronic via a user computing device (not shown), such as a client computing device 110 or 112 in FIG. 1, communicating with the healthcare cognitive system 300 via one or more data communication links and potentially one or more data networks.

[0077] As shown in FIG. 3, in accordance with one illustrative embodiment, a patient 302 presents symptoms 304 of a medical malady or condition to a user 306, such as a healthcare practitioner, technician, or the like. The user 306 may interact with the patient 302 via a question 314 and response 316 exchange where the user gathers more information about the patient 302, the symptoms 304, and the medical malady or condition of the patient 302. It should be appreciated that the questions/responses may in fact also represent the user 306 gathering information from the patient 302 using various medical equipment, e.g., blood pressure monitors, thermometers, wearable health and activity monitoring devices associated with the patient such as a FitBit™, a wearable heart monitor, or any other medical equipment that may monitor one or more medical characteristics of the patient 302. In some cases, such medical equipment may be medical equipment typically used in hospitals or medical centers to monitor vital signs and medical conditions of patients that are present in hospital beds for observation or medical treatment.

[0078] In response, the user 302 submits a request 308 to the healthcare cognitive system 300, such as via a user interface on a client computing device that is configured to allow users to submit requests to the healthcare cognitive system 300 in a format that the healthcare cognitive system 300 can parse and process. The request 308 may include, or be accompanied with, information identifying patient attributes 318. These patient attributes 318 may include, for example, an identifier of the patient 302 from which patient's EMRs 322 for the patient may be retrieved, demographic information about the patient, the symptoms 304, and other pertinent information obtained from the responses 316 to the questions 314 or information obtained from medical equipment used to monitor or gather data about the condition of the patient 302. Any information about the patient 302 that may be relevant to a cognitive evaluation of the patient by the healthcare cognitive system 300 may be included in the request 308 and/or patient attributes 318.

[0079] The healthcare cognitive system 300 provides a cognitive system that is specifically configured to perform an implementation specific healthcare oriented cognitive operation. In the depicted example, this healthcare oriented cognitive operation is directed to providing a treatment

recommendation 328 to the user 306 to assist, the user 306 in treating the patient 302 based on their reported symptoms 304 and other information gathered about the patient 302 via the question 314 and response 316 process and/or medical equipment monitoring/data gathering. The healthcare cognitive system 300 operates on the request 308 and patient attributes 318 utilizing information gathered from the medical corpus and other source data 326, treatment guidance data 324, and the patient's EMRs 322 associated with the patient 302 to generate one or more treatment recommendation 328. The treatment recommendations 328 may be presented in a ranked ordering with associated supporting evidence, obtained from the patient attributes 318 and data sources 322-326, indicating the reasoning as to why the treatment recommendation 328 is being provided and why it is ranked in the manner that it is ranked.

[0080] For example, based on the request 308 and the patient attributes 318, the healthcare cognitive system 300 may operate on the request, such as by using a QA pipeline type processing as described herein, to parse the request 308 and patient attributes 318 to determine what is being requested and the criteria upon which the request is to be generated as identified by the patient attributes 318, and may perform various operations for generating queries that are sent to the data sources 322-326 to retrieve data, generate candidate treatment recommendations (or answers to the input question), and score these candidate treatment recommendations based on supporting evidence found in the data sources 322-326. In the depicted example, the patient's EMRs 322 is a patient information repository that collects patient data from a variety of sources, e.g., hospitals, laboratories, physicians' offices, health insurance companies, pharmacies, etc. The patient's EMRs 322 store various information about individual patients, such as patient 302, in a manner (structured, unstructured, or a mix of structured and unstructured formats) that the information may be retrieved and processed by the healthcare cognitive system 300. This patient information may comprise various demographic information about patients, personal contact information about patients, employment information, health insurance information, laboratory reports, physician reports from office visits, hospital charts, historical information regarding previous diagnoses, symptoms, treatments, prescription information, etc. Based on an identifier of the patient 302, the patient's corresponding EMRs 322 from this patient repository may be retrieved by the healthcare cognitive system 300 and searched/processed to generate treatment recommendations 328.

[0081] The treatment guidance data 324 provides a knowledge base of medical knowledge that is used to identify potential treatments for a patient based on the patient's attributes 318 and historical information presented in the patient's EMRs 322. This treatment guidance data 324 may be obtained from official treatment guidelines and policies issued by medical authorities, the American Medical Association, may be obtained from widely accepted physician medical and reference texts, e.g., the Physician's Desk Reference, insurance company guidelines, or the like. The treatment guidance data 324 may be provided in any suitable form that may be ingested by the healthcare cognitive system 300 including both structured and unstructured formats.

[0082] In some cases, such treatment guidance data 324 may be provided in the form of rules that indicate the criteria

required to be present, and/or required not to be present, for the corresponding treatment to be applicable to a particular patient for treating a particular symptom or medical malady/condition. For example, the treatment guidance data 324 may comprise a treatment recommendation rule that indicates that for a treatment of Decitabine, strict criteria for the use of such a treatment is that the patient 302 is less than or equal to 60 years of age, has acute myeloid leukemia (AML), and no evidence of cardiac disease. Thus, for a patient 302 that is 59 years of age, has AML, and does not have any evidence in their patient attributes 318 or patient's EMRs indicating evidence of cardiac disease, the following conditions of the treatment rule exist:

[0083] Age 60 years=59 (MET);

[0084] Patient has AML=AML (MET); and

[0085] Cardiac Disease=false (MET)

Since all of the criteria of the treatment rule are met by the specific information about this patient 302, then the treatment of Decitabine is a candidate treatment for consideration for this patient 302. However, if the patient had been 69 years old, the first criterion would not have been met and the Decitabine treatment would not be a candidate treatment for consideration for this patient 302. Various potential treatment recommendations may be evaluated by the healthcare cognitive system 300 based on ingested treatment guidance data 324 to identify subsets of candidate treatments for further consideration by the healthcare cognitive system 300 by scoring such candidate treatments based on evidential data obtained from the patient's EMRs 322 and medical corpus and other source data 326.

[0086] For example, data mining processes may be employed to mine the data in sources 322 and 326 to identify evidential data supporting and/or refuting the applicability of the candidate treatments to the particular patient 302 as characterized by the patient's patient attributes 318 and EMRs 322. For example, for each of the criteria of the treatment rule, the results of the data mining provides a set of evidence that supports giving the treatment in the cases Where the criterion is "MET" and in cases where the criterion is "NOT MET." The healthcare cognitive system 300 processes the evidence in accordance with various cognitive logic algorithms to generate a confidence score for each candidate treatment recommendation indicating a confidence that the corresponding candidate treatment recommendation is valid for the patient 302. The candidate treatment recommendations may then be ranked according to their confidence scores and presented to the user 306 as a ranked listing of treatment recommendations 328. In some cases, only a highest ranked, or final answer, is returned as the treatment recommendation 328. The treatment recommendation 328 may be presented to the user 306 in a manner that the underlying evidence evaluated by the healthcare cognitive system 300 may be accessible, such as via a drilldown interface, so that the user 306 may identify the reasons why the treatment recommendation 328 is being provided by the healthcare cognitive system 300.

[0087] In accordance with the illustrative embodiments herein, the healthcare cognitive system 300 is augmented to include contextually relevant patient information extractor 341 for identifying portions of the patient's EMRs 322 that are most relevant to an input question 342 provided by user 306. That is, user 306 may provide the input question 342 for a specific symptom to contextually relevant patient information extractor 341, which runs input question 342 or

results of search run based on the input questions against the patient's EMRs 322 and generates search results. Patient's EMRs 322 may include unstructured natural language content and structured information content detailing at least one of encounters with a corresponding patient, procedures performed on the patient, interactions with medical practitioners concerning the patient, medications associated with the patient, results of medical tests and procedures, patient demographic information, or static medical condition information. Contextually relevant patient information extractor 341 performs post-processing on the search results to generate results 343.

[0088] In one embodiment, in running the input question 342 against the patient's EMRs 322 to generate search results 343, contextually relevant patient information extractor 341 encodes latent relationships between documents in the patient's EMRs 322. Contextually relevant patient information extractor 341 encodes the latent relationships between documents in the patient's EMRs 322 in a context graph, where each node in the context graph corresponds to a document in the patient's EMRs 322 and edges in the context graph between pairs of nodes represents the encoded latent relationships. Contextually relevant patient information extractor 341 then uses these latent relationships for identifying relevant documents. The latent relationships encoded in the context graph attempt to capture a notion of "nearness" between a pair of documents. In one embodiment, the "nearness" is a temporal nearness. That is, documents in the patient's EMRs 322 each have a timestamp associated with them, indicating a point in time that the document was created. Since these documents are all associated with the same patient, contextually relevant patient information extractor 341 uses this temporal property of each document for connecting documents that are close to each other in time within the context graph using edges.

[0089] In another embodiment, the nearness is based on specific events. That is, documents from the patient's EMRs 322 are of many different types and are generated in the context of patient care. For instance, a patient visiting a medical facility will result in an "in-patient note" and an "admission note" being created for that patient. Similarly, on discharge from the facility, a "discharge summary note" is generated. Since all these notes for the patient's stay are related to the patient's care during the same visit or event, contextually relevant patient information extractor 341 connects these notes to one another. There are other such examples of "pre-event", "during the event" and "post-event" EMR documents that contextually relevant patient information extractor 341 also links together. As an example, when the patient undergoes a procedure at a medical facility, the physician generates a "pre-operative note" detailing the reasons for the procedure and for scheduling the procedure. "Operative notes" and "progress notes" are generated during the procedure and recovery of the patient. Finally, a "discharge summary note" is generated upon the discharge of the patient from the facility. Contextually relevant patient information extractor 341 links each of these notes to one another in our context manager.

[0090] From the input question 342 provided by user 306 or results of search run based on the input questions, contextually relevant patient information extractor 341 analyzes documents in the patient's EMRs to determine entries that are "near" an initial entry found through the search of the index. For example, those documents that are temporally

near, i.e. within a predetermined time period such as within a month, a year, 5 years, or the like. The particular “nearness” definition may be different depending upon the particular candidate answer being evaluated and/or the focus or answer type of input question 342. Thus, for example, input question 342 may identify an initial entry in the patient’s EMRs 322 that mentions an irregular heartbeat, a high pulse rate, or the like (related concepts may be identified through a related concept configuration data structure used to configure the system). Contextually relevant patient information extractor 341 may identify a particular document that includes information about an irregular heartbeat, a high pulse rate, or the like, and use the node associated with the document as a starting point in the context graph from which contextually relevant patient information extractor 341 identifies other entries, documents, and the like, in the context graph that are “near” the identified node. In one illustrative embodiment, the identification performed by contextually relevant patient information extractor 341 involves a nearness, such as temporal, event related, or the like, where the boundary of the temporal nearness is defined based on the clinical factor and focus of the input question 342. Contextually relevant patient information extractor 341 then analyzes the resulting entries and documentation with regard to evidence in support of an answer to the input question 342. Thus, the corpus used to identify answers to the input question may be focused to those entries/documents that are nearest the entries initially identified as relevant to the input question 342. Contextually relevant patient information extractor 341 performs post-processing on the search results to generate an indicator value and returns the indicator value in results 343.

[0091] While FIG. 3 is depicted with an interaction between the patient 302 and a user 306, which may be a healthcare practitioner such as a physician, nurse, physician’s assistant, lab technician, or any other healthcare worker, for example, the illustrative embodiments do not require such. Rather, the patient 302 may interact directly with the healthcare cognitive system 300 without having to go through an interaction with the user 306 and the user 306 may interact with the healthcare cognitive system 300 without having to interact with the patient 302. For example, in the first case, the patient 302 may be requesting 308 treatment recommendations 328 from the healthcare cognitive system 300 directly based on the symptoms 304 provided by the patient 302 to the healthcare cognitive system 300. Moreover, the healthcare cognitive system 300 may actually have logic for automatically posing questions 314 to the patient 302 and receiving responses 316 from the patient 302 to assist with data collection for generating treatment recommendations 328. In the latter case, the user 306 may operate based on only information previously gathered and present in the patient’s EMRs 322 by sending a request 308 along with patient attributes 318 and obtaining treatment recommendations in response from the healthcare cognitive system 300. Thus, the depiction in FIG. 3 is only an example and should not be interpreted as requiring the particular interactions depicted when many modifications may be made without departing from the spirit and scope of the present invention. It should be appreciated, however, that at no time should the treatment itself be administered to the patient 302 without prior approval of the healthcare professional treating the patient, i.e. final determinations as to treatments given to a patient will always fall on the health-

care professional with the mechanisms of the illustrative embodiments serving only as an advisory tool for the healthcare professional (user 306) and/or patient 302.

[0092] As mentioned above, the healthcare cognitive system 300 may include a request processing pipeline, such as request processing pipeline 108 in FIG. 1, which may be implemented, in some illustrative embodiments, as a Question Answering (QA) pipeline. The QA pipeline may receive an input question, such as “what is the appropriate treatment for patient P?”, or a request, such as “diagnose and provide a treatment recommendation for patient P.”

[0093] FIGS. 4A-4B depict exemplary illustrations of the process performed by a contextually relevant patient information extractor, such as contextually relevant patient information extractor 341 of FIG. 3, in linking two or more documents together based on relevancy in accordance with an illustrative embodiment. In FIG. 4A, documents 402a, 402b, 402c, 402d, and 402e, which are all associated with the same patient, each have an associated timestamp. In determining a latent relationship between the documents associated with the same patient using, for example, a temporal nearness, the contextually relevant patient information extractor analyzes documents in the patient’s EMRs to determine entries that are within a predetermined time period such as, for example, a month. This analysis results in the contextually relevant patient information extractor connecting documents 402b, 402c, and 402d to each other.

[0094] In determining a latent relationship between the documents associated with the same patient using, for example, an event nearness, in FIG. 4B, the contextually relevant patient information extractor analyzes documents in the patient’s EMRs to determine entries that are associated with the patient’s visit to medical facility for a particular procedure. From the patient’s EMRs, the contextually relevant patient information extractor identifies pre-operative note 412 detailing the reasons for the patient’s procedure and for scheduling the patient’s procedure, operative note 414 generated during the patient’s procedure, progress note 416 generated during the recovery of the patient, and discharge summary note 418 detailing the discharge of the patient. Since the contextually relevant patient information extractor identifies these documents as pertaining to the same procedure, the analysis performed by the contextually relevant patient information extractor results in connecting documents 412, 414, 416, and 418 to each other.

[0095] FIG. 5 illustrates a QA pipeline of a healthcare cognitive system, such as healthcare cognitive system 300 in FIG. 3, or an implementation of cognitive system 100 in FIG. 1, for processing an input question in accordance with one illustrative embodiment. It should be appreciated that the stages of the QA pipeline shown in FIG. 5 are implemented as one or more software engines, components, or the like, which are configured with logic for implementing the functionality attributed to the particular stage. Each stage is implemented using one or more of such software engines, components or the like. The software engines, components, etc. are executed on one or more processors of one or more data processing systems or devices and utilize or operate on data stored in one or more data storage devices, memories, or the like, on one or more of the data processing systems. The QA pipeline of FIG. 5 is augmented, for example, in one or more of the stages to implement the improved mechanism of the illustrative embodiments described hereafter, additional stages may be provided to implement the improved

mechanism, or separate logic from the pipeline 500 may be provided for interfacing with the pipeline 500 and implementing the improved functionality and operations of the illustrative embodiments.

[0096] As shown in FIG. 5, the QA pipeline 500 comprises a plurality of stages 510-580 through which the cognitive system operates to analyze an input question and generate a final response. In an initial question input stage 510, the QA pipeline 500 receives an input question that is presented in a natural language format. That is, a user inputs, via a user interface, an input question for which the user wishes to obtain an answer, e.g., “What medical treatments for diabetes are applicable to a 60 year old patient with cardiac disease?” In response to receiving the input question, the next stage of the QA pipeline 500, i.e. the question and topic analysis stage 520, parses the input question using natural language processing (NLP) techniques to extract major features from the input question, and classify the major features according to types, e.g., names, dates, or any of a plethora of other defined topics. For example, in a question of the type “Who were Washington’s closest advisors?”, the term “who” may be associated with a topic for “persons” indicating that the identity of a person is being sought, “Washington” may be identified as a proper name of a person with which the question is associated, “closest” may be identified as a word indicative of proximity or relationship, and “advisors” may be indicative of a noun or other language topic. Similarly, in the previous question “medical treatments” may be associated with pharmaceuticals, medical procedures, holistic treatments, or the like, “diabetes” identifies a particular medical condition, “60 years old” indicates an age of the patient, and “cardiac disease” indicates an existing medical condition of the patient.

[0097] In addition, the extracted major features include key words and phrases, classified into question characteristics, such as the focus of the question, the lexical answer type (LAT) of the question, and the like. As referred to herein, a lexical answer type (LAT) is a word in, or a word inferred from, the input question that indicates the type of the answer, independent of assigning semantics to that word. For example, in the question “What maneuver was invented in the 1500s to speed up the game and involves two pieces of the same color?,” the LAT is the string “maneuver.” The focus of a question is the part of the question that, if replaced by the answer, makes the question a standalone statement. For example, in the question “What drug has been shown to relieve the symptoms of ADD with relatively few side effects?,” the focus is “drug” since if this word were replaced with the answer, e.g., the answer “Adderall” can be used to replace the term “drug” to generate the sentence “Adderall has been shown to relieve the symptoms of ADD with relatively few side effects.” The focus often, but not always, contains the LAT. On the other hand, in many cases it is not possible to infer a meaningful LAT from the focus.

[0098] Referring again to FIG. 5, the identified major features are then used during the question decomposition stage 530 to decompose the question into one or more queries that are applied to the corpora of data/information 545 in order to generate one or more hypotheses. The queries are generated in any known or later developed query language, such as the Structure Query Language (SQL), or the like. The queries are applied to one or more databases storing information about the electronic texts, documents, articles, websites, and the like, that make up the corpora of

data/information 545. That is, these various sources themselves, different collections of sources, and the like, represent a different corpus 547 within the corpora 545. There may be different corpora 547 defined for different collections of documents based on various criteria depending upon the particular implementation. For example, different corpora may be established for different topics, subject matter categories, sources of information, or the like. As one example, a first corpus may be associated with healthcare documents while a second corpus may be associated with financial documents. Alternatively, one corpus may be documents published by the U.S. Department of Energy while another corpus may be IBM Redbooks documents. Any collection of content having some similar attribute may be considered to be a corpus 547 within the corpora 545.

[0099] The queries are applied to one or more databases storing information about the electronic texts, documents, articles, websites, and the like, that make up the corpus of data/information, e.g., the corpus of data 106 in FIG. 1. The queries are applied to the corpus of data/information at the hypothesis generation stage 540 to generate results identifying potential hypotheses for answering the input question, which can then be evaluated. That is, the application of the queries results in the extraction of portions of the corpus of data/information matching the criteria of the particular query. These portions of the corpus are then analyzed and used, during the hypothesis generation stage 540, to generate hypotheses for answering the input question. These hypotheses are also referred to herein as “candidate answers” for the input question. For any input question, at this stage 540, there may be hundreds of hypotheses or candidate answers generated that may need to be evaluated.

[0100] The QA pipeline 500, in stage 550, then performs a deep analysis and comparison of the language of the input question and the language of each hypothesis or “candidate answer,” as well as performs evidence scoring to evaluate the likelihood that the particular hypothesis is a correct answer for the input question. As mentioned above, this involves using a plurality of reasoning algorithms, each performing a separate type of analysis of the language of the input question and/or content of the corpus that provides evidence in support of, or not in support of, the hypothesis. Each reasoning algorithm generates a score based on the analysis it performs which indicates a measure of relevance of the individual portions of the corpus of data/information extracted by application of the queries as well as a measure of the correctness of the corresponding hypothesis, i.e. a measure of confidence in the hypothesis. There are various ways of generating such scores depending upon the particular analysis being performed. In generally, however, these algorithms look for particular terms, phrases, or patterns of text that are indicative of terms, phrases, or patterns of interest and determine a degree of matching with higher degrees of matching being given relatively higher scores than lower degrees of matching.

[0101] Thus, for example, an algorithm may be configured to look for the exact term from an input question or synonyms to that term in the input question, e.g., the exact term or synonyms for the term “movie,” and generate a score based on a frequency of use of these exact terms or synonyms. In such a case, exact matches will be given the highest scores, while synonyms may be given lower scores based on a relative ranking of the synonyms as may be specified by a subject matter expert (person with knowledge

of the particular domain and terminology used) or automatically determined from frequency of use of the synonym in the corpus corresponding to the domain. Thus, for example, an exact match of the term “movie” in content of the corpus (also referred to as evidence, or evidence passages) is given a highest score. A synonym of movie, such as “motion picture” may be given a lower score but still higher than a synonym of the type “film” or “moving picture show.” Instances of the exact matches and synonyms for each evidence passage may be compiled and used in a quantitative function to generate a score for the degree of matching of the evidence passage to the input question.

[0102] Thus, for example, a hypothesis or candidate answer to the input question of “What was the first movie?” is “The Horse in Motion.” if the evidence passage contains the statements “The first motion picture ever made was ‘The Horse in Motion’ in 1878 by Eadweard Muybridge. It was a movie of a horse running,” and the algorithm is looking for exact matches or synonyms to the focus of the input question, i.e. “movie,” then an exact match of “movie” is found in the second sentence of the evidence passage and a highly scored synonym to “movie,” i.e. “motion picture,” is found in the first sentence of the evidence passage. This may be combined with further analysis of the evidence passage to identify that the text of the candidate answer is present in the evidence passage as well, i.e. “The Horse in Motion.” These factors may be combined to give this evidence passage a relatively high score as supporting evidence for the candidate answer “The Horse in Motion” being a correct answer.

[0103] It should be appreciated that this is just one simple example of how scoring can be performed. Many other algorithms of various complexities may be used to generate scores for candidate answers and evidence without departing from the spirit and scope of the present invention.

[0104] In the synthesis stage **560**, the large number of scores generated by the various reasoning algorithms are synthesized into confidence scores or confidence measures for the various hypotheses. This process involves applying weights to the various scores, where the weights have been determined through training of the statistical model employed by the QA pipeline **500** and/or dynamically updated. For example, the weights for scores generated by algorithms that identify exactly matching terms and synonym may be set relatively higher than other algorithms that are evaluating publication dates for evidence passages. The weights themselves may be specified by subject matter experts or learned through machine learning processes that evaluate the significance of characteristics evidence passages and their relative importance to overall candidate answer generation.

[0105] The weighted scores are processed in accordance with a statistical model generated through training of the QA pipeline **500** that identifies a manner by which these scores may be combined to generate a confidence score or measure for the individual hypotheses or candidate answers. This confidence score or measure summarizes the level of confidence that the QA pipeline **500** has about the evidence that the candidate answer is inferred by the input question, i.e. that the candidate answer is the correct answer for the input question.

[0106] The resulting confidence scores or measures are processed by a final confidence merging and ranking stage **570** which compares the confidence scores and measures to each other, compares them against predetermined thresh-

olds, or performs any other analysis on the confidence scores to determine which hypotheses/candidate answers are the most likely to be the correct answer to the input question. The hypotheses/candidate answers are ranked according to these comparisons to generate a ranked listing of hypotheses/candidate answers (hereafter simply referred to as “candidate answers”). From the ranked listing of candidate answers, at stage **580**, a final answer and confidence score, or final set of candidate answers and confidence scores, are generated and output to the submitter of the original input question via a graphical user interface or other mechanism for outputting information.

[0107] As shown in FIG. 5, in accordance with one illustrative embodiment, QA pipeline **500** is augmented to include contextually relevant patient information extractor **590** for identifying portions of the patient’s EMRs found in the corpora **545**. As the queries are applied to the corpora **545** of data/information at the hypothesis generation stage **540** to generate results identifying potential hypotheses for answering the input question, contextually relevant patient information extractor **590** performs post-processing on the search results obtained at the hypothesis generation stage **540**. Contextually relevant patient information extractor **590** encodes latent relationships between the identified documents in a context graph, where each node in the context graph corresponds to one of the identified documents and edges in the context graph between pairs of nodes corresponds to the encoded latent relationships. Contextually relevant patient information extractor **590** then uses these latent relationships for identifying relevant documents to capture a notion of “nearness” between a pair of document, such as temporal nearness, event nearness, or the like. Thus, contextually relevant patient information extractor **590** analyzes the identified documents from the patient’s EMRs to determine entries that are “near” an initial entry or set of initial entries found through the search of the index. The particular “nearness” definition may be different depending upon the particular candidate answer being evaluated and/or the focus or answer type.

[0108] Thus, for example, if the input question deals with an irregular heartbeat, a high pulse rate, or the like, of the patient, contextually relevant patient information extractor **590** may identify a particular document that includes information about an irregular heartbeat, a high pulse rate, or the like. Using this node as a starting point in the context graph, contextually relevant patient information extractor **590** identifies other entries, documents, and the like, in the context graph that are “near” the identified node. Contextually relevant patient information extractor **590** then analyzes the resulting entries and documentation with regard to evidence in support of the input question and returns the result to the hypothesis generation stage **540**. Thus, the corpus used to identify answers to the input question may be focused to those entries/documents that are nearest the entries initially identified as relevant to the input question.

[0109] 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.

[0110] 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.

[0111] 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.

[0112] 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 Java, 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.

[0113] 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.

[0114] 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.

[0115] 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.

[0116] FIG. 6 is a flowchart illustrating operation of a cognitive relevant patient information extractor in accordance with an illustrative embodiment. As the operation begins, the cognitive relevant patient information extractor receives an input question directed to medical information about a patient (step 602). The contextually relevant patient information extractor analyzes a patient's electronic medical records (EMRs) to identify an initial entry in the patient's EMRs corresponding to a candidate answer to the input question (step 604). The patient's EMRs are within a corpora and are related to the patient. The contextually relevant patient information extractor then analyzes a context of the patient's EMRs based on the initial entry to identify entries in the patient's EMRs that are contextually connected to the initial entry (step 606). The contextual connection to the initial entry may be one of a temporal connection, an event related connection, or the like. The contextually relevant patient information extractor then performs question-answering analysis on the initial entry and entries that are contextually connected to the initial entry to identify one or more candidate answers to the input question (step 608). Based on the question answering analysis, the contextually relevant patient information extractor outputs a final answer to the input question (step 610), with the operation ending thereafter. The operation may return to step 602 for a next input question or if further query inputs are provided.

[0117] 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.

[0118] Thus, the illustrative embodiments provide mechanisms for improving question answering system accuracy over electronic medical records (EMRs) by exploiting inter-document relationships. The mechanisms receive an input question directed to medical information about a patient. The mechanisms analyze a patient's EMRs, to identify an initial entry in the patient's EMRs corresponding to a candidate answer to the input question. The mechanisms analyze a context of the patient's EMRs based on the initial entry to identify entries in the patient's EMR that are contextually connected to the initial entry. The mechanisms perform question-answering analysis on the initial entry and entries that are contextually connected to the initial entry to identify one or more candidate answers to the input question. The mechanisms then output a final answer to the input question based on the question answering analysis.

[0119] As noted above, it should be appreciated that the illustrative embodiments may take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements. In one example embodiment, the mechanisms of the illustrative embodiments are implemented in software or program code, which includes but is not limited to firmware, resident software, microcode, etc.

[0120] A data processing system suitable for storing and/or executing program code will include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution.

[0121] Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) can be coupled to the system either directly or through intervening I/O controllers. Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modems and Ethernet cards are just a few of the currently available types of network adapters.

[0122] The description of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form 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 embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated. 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.

What is claimed is:

1. A method, in a data processing system comprising at least one processor and at least one memory, the at least one memory comprising instructions executed by the at least one processor to cause the at least one processor to implement a contextually relevant patient information extractor, wherein the method comprises:

receiving, by the contextually relevant patient information extractor, an input question directed to medical information about a patient;

analyzing, by the contextually relevant patient information extractor, a patient's electronic medical records (EMRs) to identify an initial entry in the patient's EMRs corresponding to a candidate answer to the input question;

analyzing, by the contextually relevant patient information extractor, a context of the patient's EMRs based on the initial entry to identify entries in the patient's EMR that are contextually connected to the initial entry;

performing, by the contextually relevant patient information extractor, question answering analysis on the initial entry and entries that are contextually connected to the initial entry to identify one or more candidate answers to the input question; and

outputting, by the contextually relevant patient information extractor, a final answer to the input question based on the question answering analysis.

2. The method of claim 1, wherein the context graph comprises a set of nodes and a set of edges, wherein each node in the set of nodes of the context graph corresponds to a document in the patient's EMRs and wherein each edge in the set of edges corresponds to a latent relationship.

3. The method of claim 1, wherein the contextual connection to the initial entry is a temporal connection.

4. The method of claim 1, wherein the contextual connection to the initial entry is an event related connection.

5. The method of claim 1, wherein the contextual connection to the initial entry is an event related connection within a predetermined time period.

6. The method of claim 1, wherein the patient's EMRs comprise unstructured natural language content and structured information content detailing at least one of encounters with a corresponding patient, procedures performed on the patient, interactions with medical practitioners concerning the patient, medications associated with the patient, results of medical tests and procedures, patient demographic information, or static medical condition information.

7. The method of claim 6, wherein automatically executing the query specification on the patient's EMRs further comprises:

performing, by the contextually relevant patient information extractor, natural language processing operations on the unstructured natural language content to extract the search results in accordance with the parameters specified in the query specification.

8. A computer program product comprising a computer readable storage medium having a computer readable program stored therein, wherein the computer readable program, when executed on a computing device, causes the computing device to implement a contextually relevant patient information extractor which operates to:

receive, by the contextually relevant patient information extractor, an input question directed to medical information about a patient;

analyze, by the contextually relevant patient information extractor, a patient's electronic medical records (EMRs) to identify an initial entry in the patient's EMRs corresponding to a candidate answer to the input question;

analyze, by the contextually relevant patient information extractor, a context of the patient's EMRs based on the initial entry to identify entries in the patient's EMR that are contextually connected to the initial entry;

perform, by the contextually relevant patient information extractor, question answering analysis on the initial entry and entries that are contextually connected to the initial entry to identify one or more candidate answers to the input question; and

output, by the contextually relevant patient information extractor, a final answer to the input question based on the question answering analysis.

9. The computer program product of claim 8, wherein the context graph comprises a set of nodes and a set of edges, wherein each node in the set of nodes of the context graph corresponds to a document in the patient's EMRs and wherein each edge in the set of edges corresponds to a latent relationship.

10. The computer program product of claim 8, wherein the contextual connection to the initial entry is a temporal connection.

11. The computer program product of claim 8, wherein the contextual connection to the initial entry is an event related connection.

12. The computer program product of claim 8, wherein the contextual connection to the initial entry is an event related connection within a predetermined time period.

13. The computer program product of claim 8, wherein the patient's EMRs comprise unstructured natural language content and structured information content detailing at least one of encounters with a corresponding patient, procedures performed on the patient, interactions with medical practitioners concerning the patient, medications associated with the patient, results of medical tests and procedures, patient demographic information, or static medical condition information.

14. The computer program product of claim 13, wherein the computer readable program to automatically execute the query specification on the patient's EMRs further causes the computing device to:

perform, by the contextually relevant patient information extractor, natural language processing operations on the unstructured natural language content to extract the search results in accordance with the parameters specified in the query specification.

15. An apparatus comprising:

a processor; and

a memory coupled to the processor, wherein the memory comprises instructions which, when executed by the processor, cause the processor to implement a contextually relevant patient information extractor which operates to:

receive, by the contextually relevant patient information extractor, an input question directed to medical information about a patient;

analyze, by the contextually relevant patient information extractor, a patient's electronic medical records (EMRs) to identify an initial entry in the patient's EMRs corresponding to a candidate answer to the input question;

analyze, by the contextually relevant patient information extractor, a context of the patient's EMRs based on the initial entry to identify entries in the patient's EMR that are contextually connected to the initial entry;

perform, by the contextually relevant patient information extractor, question answering analysis on the initial entry and entries that are contextually connected to the initial entry to identify one or more candidate answers to the input question; and

output, by the contextually relevant patient information extractor, a final answer to the input question based on the question answering analysis.

16. The apparatus of claim 15, wherein the context graph comprises a set of nodes and a set of edges, wherein each node in the set of nodes of the context graph corresponds to a document in the patient's EMRs and wherein each edge in the set of edges corresponds to a latent relationship.

17. The apparatus of claim 15, wherein the contextual connection to the initial entry is a temporal connection.

18. The apparatus of claim 15, wherein the contextual connection to the initial entry is an event related connection.

19. The apparatus of claim 15, wherein the contextual connection to the initial entry is an event related connection within a predetermined time period.

20. The apparatus of claim 15, wherein the patient's EMRs comprise unstructured natural language content and structured information content detailing at least one of encounters with a corresponding patient, procedures performed on the patient, interactions with medical practitioners concerning the patient, medications associated with the patient, results of medical tests and procedures, patient demographic information, or static medical condition information.

* * * * *