



(51) International Patent Classification:

G06F 40/284 (2020.01) G16H 20/00 (2018.01)
G06F 40/205 (2020.01) G06N 3/096 (2023.01)
G06F 18/21 (2023.01) G06F 11/00 (2006.01)
G16H 10/60 (2018.01)

(21) International Application Number:

PCT/US2024/027050

(22) International Filing Date:

30 April 2024 (30.04.2024)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

63/500,664	08 May 2023 (08.05.2023)	US
63/522,731	23 June 2023 (23.06.2023)	US
18/649,145	29 April 2024 (29.04.2024)	US

(71) Applicant: NEC LABORATORIES AMERICA, INC.

[US/US]; 4 Independence Way, Suite 200, Princeton, NJ 08540 (US).

(72) Inventors: ZHAO, Xujiang; 2 Pittenger Road, Hillsbor-

ough, NJ 08844 (US). CHEN, Haifeng; 11 Blackhawk Court, West Windsor, NJ 08550 (US). CHENG, Wei; 5 Arnold Drive, Princeton Junction, NJ 08550 (US). LIU,

Yanchi; 34 Beechwood Court, Monmouth Junction, NJ 08852 (US). CHEN, Zhengzhang; 44 York Road, Princeton Junction, NJ 08550 (US). WANG, Haoyu; 512 Pheasant Hollow Drive, Plainsboro, NJ 08536 (US).

(74) Agent: BITETTO, James J.; 401 Broadhollow Road, Suite 402, Melville, NY 11747 (US).

(81) Designated States (unless otherwise indicated, for every

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

(84) Designated States (unless otherwise indicated, for every

kind of regional protection available): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT,

(54) Title: INFORMATION EXTRACTION WITH LARGE LANGUAGE MODELS

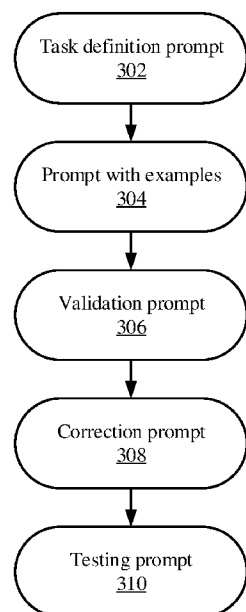


FIG. 3

(57) Abstract: Methods and systems for information extraction include configuring (304) a language model with an information extraction instruction prompt and at least one labeled example prompt. Configuration of the language model is validated (306) using at least one validation prompt. Errors made by the language model in response to the at least one validation prompt are corrected (308) using a correction prompt. Information extraction is performed (310) on an unlabeled sentence using the language model to identify a relation from the unlabeled sentence. An action is performed (210) responsive to the identified relation.

LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE,
SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN,
GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

INFORMATION EXTRACTION WITH LARGE LANGUAGE MODELS

RELATED APPLICATION INFORMATION

[0001] This application claims priority to U.S. Patent Application No. 63/500,664, filed on May 8, 2023, to U.S. Patent Application No. 63/522,731, filed on June 23, 2023, and to U.S. Patent Application No. 18/649,145 filed on April 29, 2024, each incorporated herein by reference in their entirety.

BACKGROUND

Technical Field

[0002] The present invention relates to information extraction and, more particularly, to the use of large language models to extract information from text.

Description of the Related Art

[0003] Information extraction is a natural language task where unstructured text is analyzed to identify types of information that it contains. For example, information extraction may identify named entities within the input text and may further identify relationships between such named entities.

SUMMARY

[0004] A method for information extraction includes configuring a language model with an information extraction instruction prompt and at least one labeled example prompt. Configuration of the language model is validated using at least one validation prompt. Errors made by the language model in response to the at least one validation prompt are corrected using a correction prompt. Information extraction is performed on

an unlabeled sentence using the language model to identify a relation from the unlabeled sentence. An action is performed responsive to the identified relation.

[0005] A system for information extraction includes a hardware processor and a memory that stores a computer program. When executed by the hardware processor, the computer program causes the hardware processor to configure a language model with an information extraction instruction prompt and at least one labeled example prompt, to validate configuration of the language model using at least one validation prompt, to correct errors made by the language model in response to the at least one validation prompt using a correction prompt, to perform information extraction on an unlabeled sentence using the language model to identify a relation from the unlabeled sentence, and to perform an action responsive to the identified relation.

[0006] These and other features and advantages will become apparent from the following detailed description of illustrative embodiments thereof, which is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF DRAWINGS

[0007] The disclosure will provide details in the following description of preferred embodiments with reference to the following figures wherein:

[0008] FIG. 1 is a block/flow diagram of an information extraction task that is performed on a natural language input by a large language mode, in accordance with an embodiment of the present invention;

[0009] FIG. 2 is a block/flow diagram of an information extraction task that is performed on a natural language input by a large language mode, in accordance with an embodiment of the present invention;

[0010] FIG. 3 is a block/flow diagram of a series of prompts to a large language model that configure the large language model to perform domain-specific information extraction, in accordance with an embodiment of the present invention;

[0011] FIG. 4 is a block diagram showing the use of information extraction in the context of medical decision making in a healthcare facility, in accordance with an embodiment of the present invention;

[0012] FIG. 5 is a block diagram of a computing device that can configure and use a large language model to perform information extraction, in accordance with an embodiment of the present invention;

[0013] FIG. 6 is a block diagram showing an exemplary neural network architecture that can be used as part of the large language model, in accordance with an embodiment of the present invention; and

[0014] FIG. 7 is a block diagram showing an exemplary deep neural network architecture that can be used as part of the large language model, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0015] Large language models can be used to perform a wide variety of natural language tasks, including information extraction. However, using a general purpose language model to perform a specific task can give rise to inaccuracies, as the language model is not specifically trained to perform the task. Similarly the language model may not have access to domain-specific information, reducing accuracy for domain-specific inputs. Furthermore, the construction of the prompt for the model has a large effect on the quality of the output.

[0016] To that end, specific-domain open information extraction using a large language model may be enhanced by in-context learning. Given an input sentence, the most important relations may be extracted, including a subject, an action, and an object. A set of domain-specific sentences with ground-truth labels and an associated confident label may be used, with the ground-truth labels including a subject, action, and object. An initial prompt may be designed for a general purpose language model, including a description of the information extraction task. A series of prompts may be provided some of with the domain-specific sentences and their associated labels to provide domain-specific knowledge to the language model. Additional prompts may be used to test the understanding of the language model and to provide corrections when the model generates erroneous results. The language model can then be used to extract information from unlabeled new sentences and to provide a confidence score as to the correctness of its output.

[0017] Referring now to FIG. 1, an exemplary information extraction task is shown. Input text 102 is provided to a large language model 104. The input sentences 102 are made part of one or more prompts, which instruct the large language model 104 to perform information extraction. In this example, the information extraction task specifically extracts subjects, actions, and objects and generates output 106, but it should be understood that any appropriate information extraction task may be performed instead. For example, some embodiments may further extract adverbs.

[0018] Any appropriate language model can be used for the large language model 104. Exemplary language models include ChatGPT, GPT3, and GPT4. The large language model may be implemented as a machine learning model, such as a transformer neural network model, having parameters that are pretrained on a large corpus of textual material.

[0019] Referring now to FIG. 2, a method for performing information extraction with a large language model is shown. Block 202 applies training prompts to the language model 104, defining the information extraction task and providing examples from a set of labeled training data. The training data may pertain to a particular domain that was not used in the original training of the language model 104.

[0020] Block 204 may then perform error correction, for example by prompting the language model 104 to perform information extraction on additional examples from the labeled training data and comparing its predictions to the predetermined labels. The error correction of block 204 may include further prompts to the language model 104 instructing it as to the correct answer in instances when performs the task inaccurately.

[0021] Once the language model 104 has been properly instructed as to the task, block 206 performs information extraction on a new sentence input and block 208 determines a confidence associated with the extracted information. The confidence score may be provided by a human annotator, such as a subject-matter expert. Based on the output of the language model 104, block 210 may perform a responsive action. For example, this action may take place in the context of a medical treatment at a healthcare facility, as described in greater detail below. The responsive action may be made conditional on a threshold level of confidence, such as at least 90%.

[0022] For a set of training examples x , there may be a set of sentences $S = \{S_1, S_2, \dots, S_N\}$, where N is the number of sentences in the set, and a set of respective ground truth labels $Y = \{y_1, \dots, y_N\}$, where each $y_n = \{(s_n^1, a_n^1, o_n^1), \dots, (s_n^K, a_n^K, o_n^K)\}$, where s_n^k represents a subject, a_n^k represents an action, o_n^k represents an object, and K is the number of relations for the sentence S_n .

[0023] The training prompts include a definition of the role for the system. For example, this may be expressed in natural language as a command or instruction, such

as, “You are an AI assistant for open information extraction.” Subsequent training prompts provide in-context learning, for example defining a format of the relations that are to be extracted. For example, such a training prompt may be expressed as:

[0024] “Identify all combinations of subject, action, and object for any given sentences, and present them in the form of triplets: (subject, action, object). Here are m examples:

[0025] Sentence: ‘domain specific sentence: S_1 ’

[0026] Relations:

[0027] 1. (s_1^1, a_1^1, o_1^1)

[0028] 2. (s_1^2, a_1^2, o_1^2)

[0029] ...

[0030] K. (s_1^K, a_1^K, o_1^K)

[0031] Sentence: ‘domain specific sentence: S_2 ’

[0032] Relations:

[0033] 1. (s_2^1, a_2^1, o_2^1)

[0034] 2. (s_2^2, a_2^2, o_2^2)

[0035] ...

[0036] K. (s_2^K, a_2^K, o_2^K)

[0037] ...

[0038] Sentence: ‘domain specific sentence: S_m ’

[0039] Relations:

[0040] 1. (s_m^1, a_m^1, o_m^1)

[0041] 2. (s_m^2, a_m^2, o_m^2)

[0042] ...

[0043] K. (s_m^K, a_m^K, o_m^K)

[0044] Now please identify all combinations of subject, action, and object for the m sentences.”

[0045] Based on this in-context learning prompt, the language model 104 learns domain-specific knowledge and predicts open information for an input sentence. Thus for an input sentence p , the extracted information may be denoted herein as r_p .

[0046] The error correction of block 204 helps the language model 104 adapt to the domain of the training data. Given training examples $\{S, Y\}$, the language model 104 predicts open information contained therein. Given a sentence s_n , the language model prediction r_p is a set of extracted relations. An error correction prompt may be generated, for example as:

[0047] “No, the correct combinations of relations are:

[0048] 1. (s_m^1, a_m^1, o_{M1}^1)

[0049] 2. (s_m^2, a_m^2, o_m^2)

[0050] ...

[0051] K. (s_m^K, a_m^K, o_m^K) ”

[0052] Based on the error correction prompt, the large language model is able to generate higher-accuracy predictions for information extraction, denoted here in as $\hat{r}_p$.

[0053] Blocks 202 and 204 may further provide instructions to the language model 104 regarding prediction of confidence scores. For example, a prompt may be constructed as:

[0054] “For this sentence p , identify all combinations of subject, action, and object for any given sentences, and also provide the confidence for each combination.”

[0055] This prompt generates a confidence score c along with the prediction r_p . The confidence score may be defined as $c = [c_1, \dots, c_k]$ and the ground truth confidence

may be represented as $\hat{c}$. A further prompt may then be issued to perform error correction, for example as:

[0056] “No, your confidence is not correct. The correct confidence is $\hat{c}$.”

[0057] After the confidence correction, the language model 104 can provide its confidence prediction via a further prompt:

[0058] “For this sentence p , please provide the confidence of the relation r_p .”

[0059] Referring now to FIG. 3, a sequence of prompts is shown. The first prompt is a task definition prompt 302, which instructs the language model 104 as to what type of information is being extracted. This is followed by a prompt with a set of examples drawn from the training data. These two prompts provide information to the language model 104 as to how to respond to subsequent prompts and further provide contextual information regarding any domain-specific knowledge that may be encoded in the examples.

[0060] The initial setup is followed by a validation prompt 306, which draws examples from the labeled training data and asks the primed language model 104 to predict the predetermined labels for those examples. For any examples where the language model 104 provides incorrect answers, a correction prompt 308 is used to identify what example the language model 104 predicted incorrectly and what the correct answers should have been.

[0061] After the correction has been performed, additional validation prompts 306 may be used to ensure the language model 104 is operating correctly. If so, a testing prompt 310 with a new sentence or sentence may be applied to the language model 104 to extract information and to generate a confidence score.

[0062] Referring now to FIG. 4, a diagram of information extraction is shown in the context of a healthcare facility 400. Information extraction 408 may be used to process

patient information, for example by identifying relevant facts from patient history information and medical records 406. The information extraction 408 may be implemented with a large language model using a set of domain-specific training examples that are labeled in accordance with their usage in a medical context. A large amount of patient information can be rapidly processed in this manner to identify important information, such as exposure to allergens, medications taken, and the circumstances of an injury.

[0063] The healthcare facility may include one or more medical professionals 402 who review information extracted from a patient's medical records 406 to determine their healthcare and treatment needs. These medical records 406 may include self-reported information from the patient, test results, and notes by healthcare personnel made to the patient's file. Treatment systems 404 may furthermore monitor patient status to generate medical records 406 and may be designed to automatically administer and adjust treatments as needed.

[0064] Based on information drawn from the information extraction 408, the medical professionals 402 may then make medical decisions about patient healthcare suited to the patient's needs. For example, the medical professionals 402 may make a diagnosis of the patient's health condition and may prescribe particular medications, surgeries, and/or therapies.

[0065] The different elements of the healthcare facility 400 may communicate with one another via a network 410, for example using any appropriate wired or wireless communications protocol and medium. Thus information extraction 408 receives textual information from medical professionals 402, from treatment systems 404, from medical records 406, and updates the medical records 406 with the output of the language model, and may coordinate with treatment systems 404 in some cases to

automatically administer or alter a treatment. For example, if the information extraction 408 indicates a potential allergic reaction to a treatment or medication, the treatment systems 404 may automatically halt the administration of the treatment.

[0066] Referring now to FIG. 5, an exemplary computing device 500 is shown, in accordance with an embodiment of the present invention. The computing device 500 is configured to perform tissue analysis.

[0067] The computing device 500 may be embodied as any type of computation or computer device capable of performing the functions described herein, including, without limitation, a computer, a server, a rack based server, a blade server, a workstation, a desktop computer, a laptop computer, a notebook computer, a tablet computer, a mobile computing device, a wearable computing device, a network appliance, a web appliance, a distributed computing system, a processor-based system, and/or a consumer electronic device. Additionally or alternatively, the computing device 500 may be embodied as one or more compute sleds, memory sleds, or other racks, sleds, computing chassis, or other components of a physically disaggregated computing device.

[0068] As shown in FIG. 5, the computing device 500 illustratively includes the processor 510, an input/output subsystem 520, a memory 530, a data storage device 540, and a communication subsystem 550, and/or other components and devices commonly found in a server or similar computing device. The computing device 500 may include other or additional components, such as those commonly found in a server computer (e.g., various input/output devices), in other embodiments. Additionally, in some embodiments, one or more of the illustrative components may be incorporated in, or otherwise form a portion of, another component. For example, the memory 530, or portions thereof, may be incorporated in the processor 510 in some embodiments.

[0069] The processor 510 may be embodied as any type of processor capable of performing the functions described herein. The processor 510 may be embodied as a single processor, multiple processors, a Central Processing Unit(s) (CPU(s)), a Graphics Processing Unit(s) (GPU(s)), a single or multi-core processor(s), a digital signal processor(s), a microcontroller(s), or other processor(s) or processing/controlling circuit(s).

[0070] The memory 530 may be embodied as any type of volatile or non-volatile memory or data storage capable of performing the functions described herein. In operation, the memory 530 may store various data and software used during operation of the computing device 500, such as operating systems, applications, programs, libraries, and drivers. The memory 530 is communicatively coupled to the processor 510 via the I/O subsystem 520, which may be embodied as circuitry and/or components to facilitate input/output operations with the processor 510, the memory 530, and other components of the computing device 500. For example, the I/O subsystem 520 may be embodied as, or otherwise include, memory controller hubs, input/output control hubs, platform controller hubs, integrated control circuitry, firmware devices, communication links (e.g., point-to-point links, bus links, wires, cables, light guides, printed circuit board traces, etc.), and/or other components and subsystems to facilitate the input/output operations. In some embodiments, the I/O subsystem 520 may form a portion of a system-on-a-chip (SOC) and be incorporated, along with the processor 510, the memory 530, and other components of the computing device 500, on a single integrated circuit chip.

[0071] The data storage device 540 may be embodied as any type of device or devices configured for short-term or long-term storage of data such as, for example, memory devices and circuits, memory cards, hard disk drives, solid state drives, or other

data storage devices. The data storage device 540 can store program code 540A for configuring a model, 540B for performing information extraction, and/or 540C for performing diagnosis and treatment. Any or all of these program code blocks may be included in a given computing system. The communication subsystem 550 of the computing device 500 may be embodied as any network interface controller or other communication circuit, device, or collection thereof, capable of enabling communications between the computing device 500 and other remote devices over a network. The communication subsystem 550 may be configured to use any one or more communication technology (e.g., wired or wireless communications) and associated protocols (e.g., Ethernet, InfiniBand®, Bluetooth®, Wi-Fi®, WiMAX, etc.) to effect such communication.

[0072] As shown, the computing device 500 may also include one or more peripheral devices 560. The peripheral devices 560 may include any number of additional input/output devices, interface devices, and/or other peripheral devices. For example, in some embodiments, the peripheral devices 560 may include a display, touch screen, graphics circuitry, keyboard, mouse, speaker system, microphone, network interface, and/or other input/output devices, interface devices, and/or peripheral devices.

[0073] Of course, the computing device 500 may also include other elements (not shown), as readily contemplated by one of skill in the art, as well as omit certain elements. For example, various other sensors, input devices, and/or output devices can be included in computing device 500, depending upon the particular implementation of the same, as readily understood by one of ordinary skill in the art. For example, various types of wireless and/or wired input and/or output devices can be used. Moreover, additional processors, controllers, memories, and so forth, in various configurations can also be utilized. These and other variations of the processing system 500 are readily

contemplated by one of ordinary skill in the art given the teachings of the present invention provided herein.

[0074] Referring now to FIGs. 6 and 7, exemplary neural network architectures are shown, which may be used to implement parts of the present models, such as the large language model 104. A neural network is a generalized system that improves its functioning and accuracy through exposure to additional empirical data. The neural network becomes trained by exposure to the empirical data. During training, the neural network stores and adjusts a plurality of weights that are applied to the incoming empirical data. By applying the adjusted weights to the data, the data can be identified as belonging to a particular predefined class from a set of classes or a probability that the input data belongs to each of the classes can be output.

[0075] The empirical data, also known as training data, from a set of examples can be formatted as a string of values and fed into the input of the neural network. Each example may be associated with a known result or output. Each example can be represented as a pair, (x, y), where x represents the input data and y represents the known output. The input data may include a variety of different data types, and may include multiple distinct values. The network can have one input node for each value making up the example's input data, and a separate weight can be applied to each input value. The input data can, for example, be formatted as a vector, an array, or a string depending on the architecture of the neural network being constructed and trained.

[0076] The neural network "learns" by comparing the neural network output generated from the input data to the known values of the examples, and adjusting the stored weights to minimize the differences between the output values and the known values. The adjustments may be made to the stored weights through back propagation, where the effect of the weights on the output values may be determined by calculating

the mathematical gradient and adjusting the weights in a manner that shifts the output towards a minimum difference. This optimization, referred to as a gradient descent approach, is a non-limiting example of how training may be performed. A subset of examples with known values that were not used for training can be used to test and validate the accuracy of the neural network.

[0077] During operation, the trained neural network can be used on new data that was not previously used in training or validation through generalization. The adjusted weights of the neural network can be applied to the new data, where the weights estimate a function developed from the training examples. The parameters of the estimated function which are captured by the weights are based on statistical inference.

[0078] In layered neural networks, nodes are arranged in the form of layers. An exemplary simple neural network has an input layer 620 of source nodes 622, and a single computation layer 630 having one or more computation nodes 632 that also act as output nodes, where there is a single computation node 632 for each possible category into which the input example could be classified. An input layer 620 can have a number of source nodes 622 equal to the number of data values 612 in the input data 610. The data values 612 in the input data 610 can be represented as a column vector. Each computation node 632 in the computation layer 630 generates a linear combination of weighted values from the input data 610 fed into input nodes 620, and applies a non-linear activation function that is differentiable to the sum. The exemplary simple neural network can perform classification on linearly separable examples (e.g., patterns).

[0079] A deep neural network, such as a multilayer perceptron, can have an input layer 620 of source nodes 622, one or more computation layer(s) 630 having one or more computation nodes 632, and an output layer 640, where there is a single output

node 642 for each possible category into which the input example could be classified. An input layer 620 can have a number of source nodes 622 equal to the number of data values 612 in the input data 610. The computation nodes 632 in the computation layer(s) 630 can also be referred to as hidden layers, because they are between the source nodes 622 and output node(s) 642 and are not directly observed. Each node 632, 642 in a computation layer generates a linear combination of weighted values from the values output from the nodes in a previous layer, and applies a non-linear activation function that is differentiable over the range of the linear combination. The weights applied to the value from each previous node can be denoted, for example, by $w_1, w_2, \dots, w_{n-1}, w_n$. The output layer provides the overall response of the network to the input data. A deep neural network can be fully connected, where each node in a computational layer is connected to all other nodes in the previous layer, or may have other configurations of connections between layers. If links between nodes are missing, the network is referred to as partially connected.

[0080] Training a deep neural network can involve two phases, a forward phase where the weights of each node are fixed and the input propagates through the network, and a backwards phase where an error value is propagated backwards through the network and weight values are updated.

[0081] The computation nodes 632 in the one or more computation (hidden) layer(s) 630 perform a nonlinear transformation on the input data 612 that generates a feature space. The classes or categories may be more easily separated in the feature space than in the original data space.

[0082] Embodiments described herein may be entirely hardware, entirely software or including both hardware and software elements. In a preferred embodiment, the

present invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

[0083] Embodiments may include a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. A computer-usable or computer readable medium may include any apparatus that stores, communicates, propagates, or transports the program for use by or in connection with the instruction execution system, apparatus, or device. The medium can be magnetic, optical, electronic, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. The medium may include a computer-readable storage medium such as a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk, etc.

[0084] Each computer program may be tangibly stored in a machine-readable storage media or device (e.g., program memory or magnetic disk) readable by a general or special purpose programmable computer, for configuring and controlling operation of a computer when the storage media or device is read by the computer to perform the procedures described herein. The inventive system may also be considered to be embodied in a computer-readable storage medium, configured with a computer program, where the storage medium so configured causes a computer to operate in a specific and predefined manner to perform the functions described herein.

[0085] A data processing system suitable for storing and/or executing program code may 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 to reduce the number of times code is retrieved from bulk storage during execution. Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) may be coupled to the system either directly or through intervening I/O controllers.

[0086] 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 modem and Ethernet cards are just a few of the currently available types of network adapters.

[0087] As employed herein, the term “hardware processor subsystem” or “hardware processor” can refer to a processor, memory, software or combinations thereof that cooperate to perform one or more specific tasks. In useful embodiments, the hardware processor subsystem can include one or more data processing elements (e.g., logic circuits, processing circuits, instruction execution devices, etc.). The one or more data processing elements can be included in a central processing unit, a graphics processing unit, and/or a separate processor- or computing element-based controller (e.g., logic gates, etc.). The hardware processor subsystem can include one or more on-board memories (e.g., caches, dedicated memory arrays, read only memory, etc.). In some embodiments, the hardware processor subsystem can include one or more memories that can be on or off board or that can be dedicated for use by the hardware processor subsystem (e.g., ROM, RAM, basic input/output system (BIOS), etc.).

[0088] In some embodiments, the hardware processor subsystem can include and execute one or more software elements. The one or more software elements can include an operating system and/or one or more applications and/or specific code to achieve a specified result.

[0089] In other embodiments, the hardware processor subsystem can include dedicated, specialized circuitry that performs one or more electronic processing functions to achieve a specified result. Such circuitry can include one or more application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), and/or programmable logic arrays (PLAs).

[0090] These and other variations of a hardware processor subsystem are also contemplated in accordance with embodiments of the present invention.

[0091] Reference in the specification to “one embodiment” or “an embodiment” of the present invention, as well as other variations thereof, means that a particular feature, structure, characteristic, and so forth described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, the appearances of the phrase “in one embodiment” or “in an embodiment”, as well as any other variations, appearing in various places throughout the specification are not necessarily all referring to the same embodiment. However, it is to be appreciated that features of one or more embodiments can be combined given the teachings of the present invention provided herein.

[0092] It is to be appreciated that the use of any of the following “/”, “and/or”, and “at least one of”, for example, in the cases of “A/B”, “A and/or B” and “at least one of A and B”, is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further example, in the cases of “A, B, and/or C” and “at least one of A, B, and C”, such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C)

only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended for as many items listed.

[0093] The foregoing is to be understood as being in every respect illustrative and exemplary, but not restrictive, and the scope of the invention disclosed herein is not to be determined from the Detailed Description, but rather from the claims as interpreted according to the full breadth permitted by the patent laws. It is to be understood that the embodiments shown and described herein are only illustrative of the present invention and that those skilled in the art may implement various modifications without departing from the scope and spirit of the invention. Those skilled in the art could implement various other feature combinations without departing from the scope and spirit of the invention. Having thus described aspects of the invention, with the details and particularity required by the patent laws, what is claimed and desired protected by Letters Patent is set forth in the appended claims.

WHAT IS CLAIMED IS:

1. A computer-implemented method for information extraction, comprising:
configuring (304) a language model with an information extraction instruction prompt and at least one labeled example prompt;
validating (306) configuration of the language model using at least one validation prompt;
correcting (308) errors made by the language model in response to the at least one validation prompt using a correction prompt;
performing (310) information extraction on an unlabeled sentence using the language model to identify a relation from the unlabeled sentence; and
performing (210) an action responsive to the identified relation.
2. The method of claim 1, wherein the information extraction instruction prompt and the at least one labeled example prompt include a textual description of an information extraction task, including a definition of a relation format.
3. The method of claim 1, wherein the at least one labeled example prompt is drawn from a set of training data that includes sentences and associated relations.
4. The method of claim 3, wherein the at least one validation prompt is also drawn from the set of training data.
5. The method of claim 4, wherein the correction prompt identifies a response to the at least validation prompt that does not match a label of the at least one validation prompt from the training data and provides supplies the language model with the label.

6. The method of claim 1, wherein the at least one labeled example prompt includes a confidence score and wherein inputting the test prompt to the language model further determines a confidence score associated with the relation.

7. The method of claim 1, wherein the unlabeled sentence relates to a patient's medical condition.

8. The method of claim 7, wherein performing the action includes automatically adjusting a patient's treatment based on the identified relation.

9. The method of claim 7, wherein the identified relation is stored in a medical history of the patient to aid in medical decision making by a healthcare professional.

10. The method of claim 1, wherein the language model is a pretrained large language model based on a machine learning model.

11. A system for information extraction, comprising:
a hardware processor (510); and
a memory (540) that stores a computer program which, when executed by the hardware processor, causes the hardware processor to:

configure (304) a language model with an information extraction instruction prompt and at least one labeled example prompt;

validate (306) configuration of the language model using at least one validation prompt;

correct (308) errors made by the language model in response to the at least one validation prompt using a correction prompt;

perform (310) information extraction on an unlabeled sentence using the language model to identify a relation from the unlabeled sentence; and

perform (210) an action responsive to the identified relation.

12. The system of claim 11, wherein the information extraction instruction prompt and the at least one labeled example prompt include a textual description of an information extraction task, including a definition of a relation format.

13. The system of claim 11, wherein the at least one labeled example prompt is drawn from a set of training data that includes sentences and associated relations.

14. The system of claim 13, wherein the at least one validation prompt is also drawn from the set of training data.

15. The system of claim 14, wherein the correction prompt identifies a response to the at least validation prompt that does not match a label of the at least one validation prompt from the training data and provides supplies the language model with the label.

16. The system of claim 11, wherein the at least one labeled example prompt includes a confidence score and wherein inputting the test prompt to the language model further determines a confidence score associated with the relation.

17. The system of claim 11, wherein the unlabeled sentence relates to a patient's medical condition.

18. The system of claim 17, wherein the computer program further causes the hardware processor to automatically adjust a patient's treatment based on the identified relation.

19. The system of claim 17, wherein the identified relation is stored in a medical history of the patient to aid in medical decision making by a healthcare professional.

20. The system of claim 11, wherein the language model is a pretrained large language model based on a machine learning model.

1/6

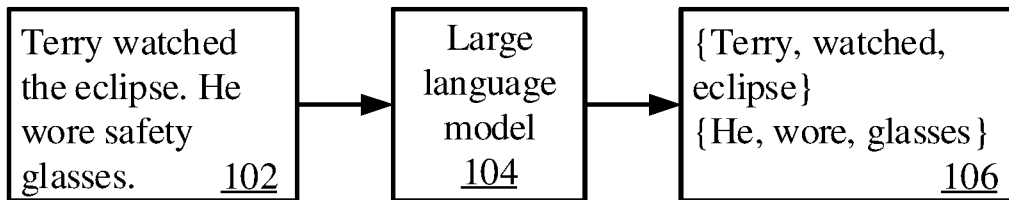


FIG. 1

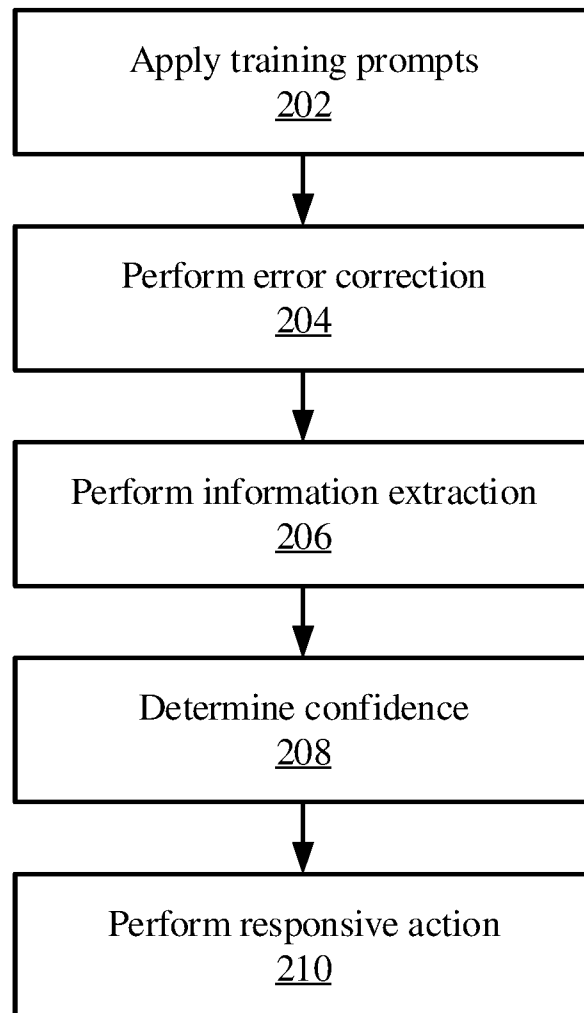


FIG. 2

2/6

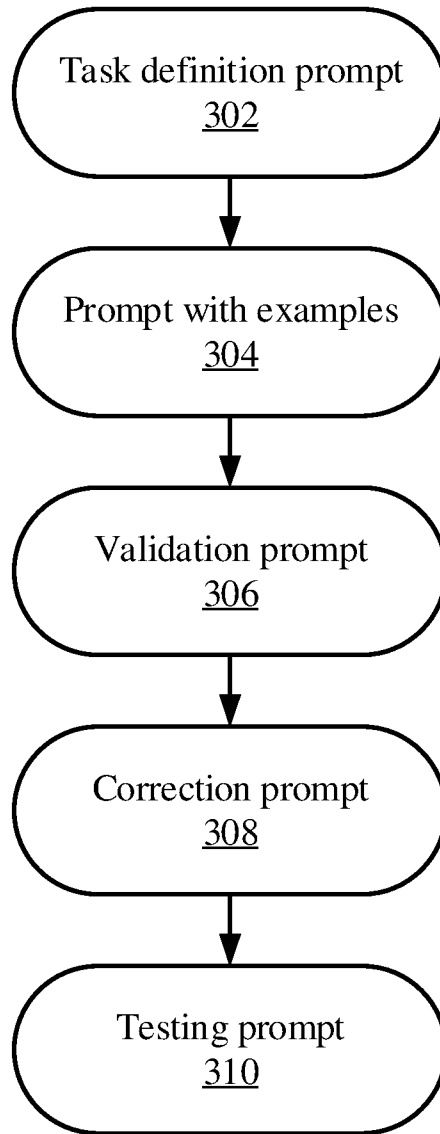


FIG. 3

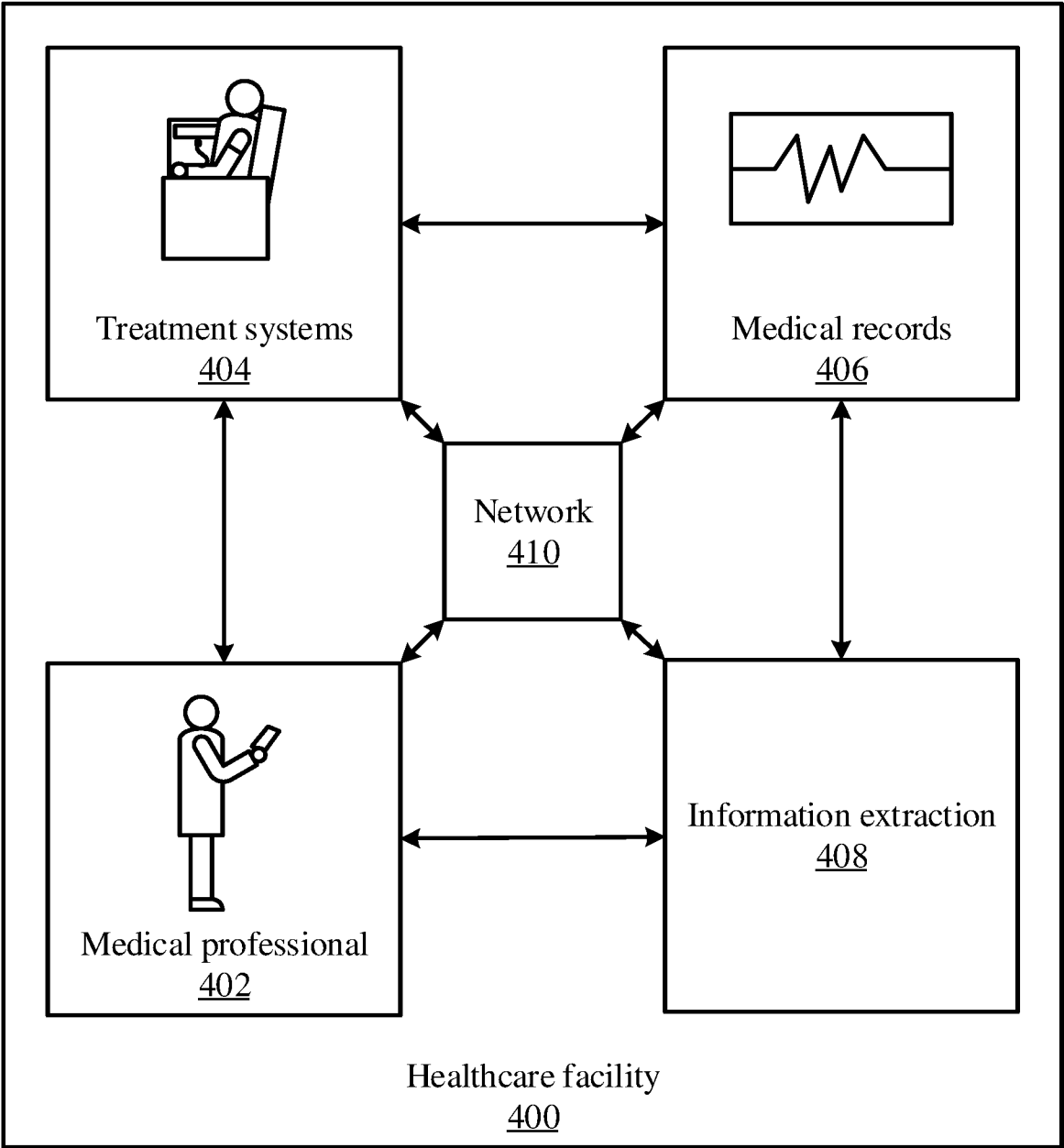


FIG. 4

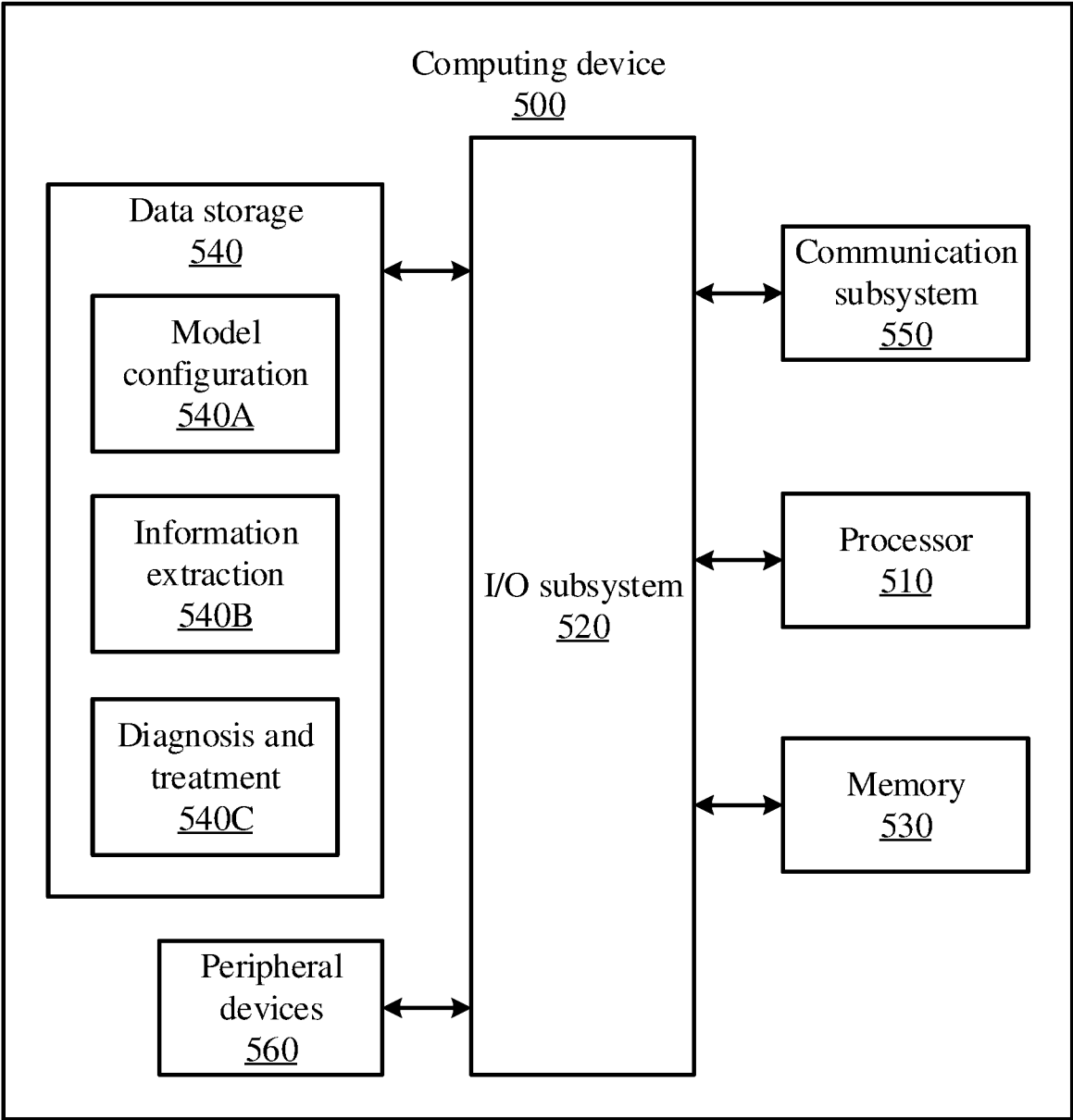


FIG. 5

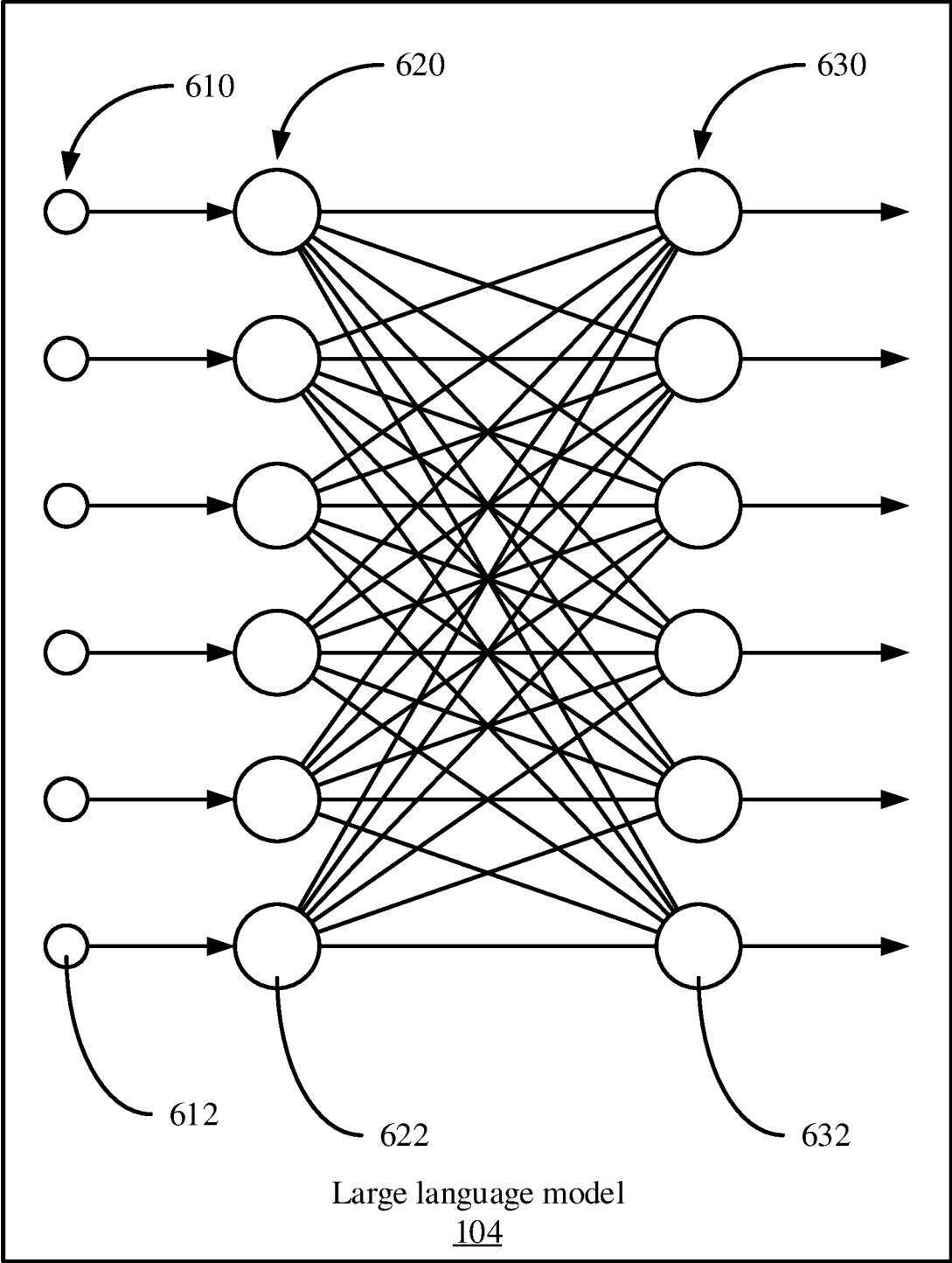


FIG. 6

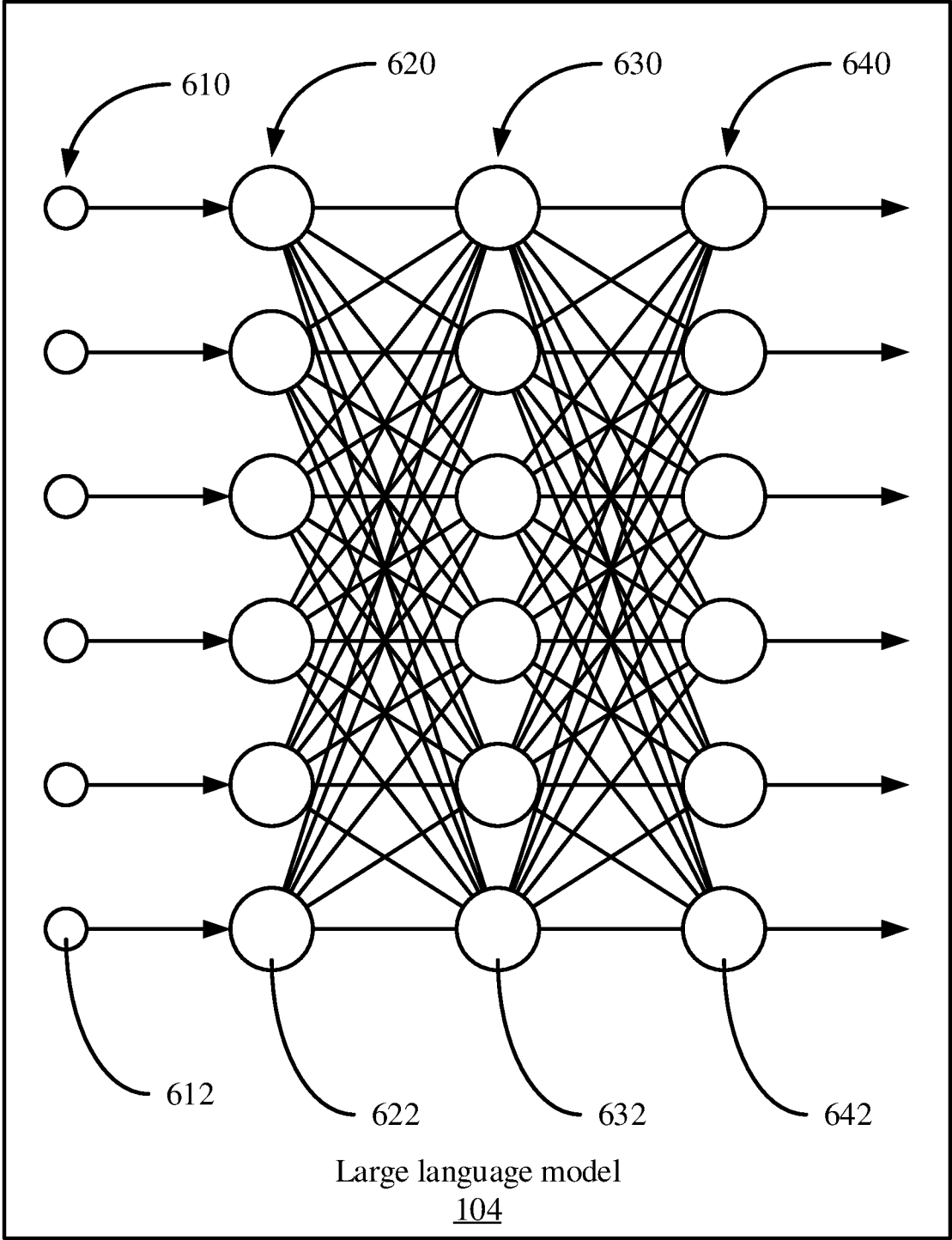


FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2024/027050

A. CLASSIFICATION OF SUBJECT MATTER G06F 40/284 (2020.01)i; G06F 40/205 (2020.01)i; G06F 18/21 (2023.01)i; G16H 10/60 (2018.01)i; G16H 20/00 (2018.01)i; G06N 3/096 (2023.01)i; G06F 11/00 (2006.01)i According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06F 40/284(2020.01); G06F 16/35(2019.01); G06F 17/27(2006.01); G06F 18/214(2023.01); G06F 40/289(2020.01); G06F 40/295(2020.01); G06K 9/62(2006.01) Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords: language model, validation, prompt, correct, error, unlabeled sentence		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SHREYAS SUNDARA RAMAN et al., 'Planning With Large Language Models Via Corrective Re-Prompting', NeurIPS 2022, pp. 1-21, 06 May 2023 [retrieved on 2024.07.30]. Retrieved from <https://openreview.net/forum?id=cMDMRBe1TKs> pages 1-10	1-20
A	US 2011-0270604 A1 (YANJUN QI et al.) 03 November 2011 (2011-11-03) paragraphs [0016]-[0096]; claims 1-25; and figures 1, 5	1-20
A	CN 115687609 A (BEIJING JIAOTONG UNIVERSITY) 03 February 2023 (2023-02-03) paragraphs [0040]-[0070]; claims 1-2; and figures 1-2	1-20
A	CN 115017312 A (ALIBABA (CHINA) CO., LTD.) 06 September 2022 (2022-09-06) paragraphs [0066]-[0134]; claims 1-6; and figures 1-7	1-20
A	CN 115640808 A (SUZHOU INSPUR INTELLIGENT TECHNOLOGY CO., LTD.) 24 January 2023 (2023-01-24) paragraphs [0036]-[0129]; claims 1-17; and figures 1-5	1-20
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents: “A” document defining the general state of the art which is not considered to be of particular relevance “D” document cited by the applicant in the international application “E” earlier application or patent but published on or after the international filing date “L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) “O” document referring to an oral disclosure, use, exhibition or other means “P” document published prior to the international filing date but later than the priority date claimed “T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention “X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone “Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art “&” document member of the same patent family		
Date of the actual completion of the international search 19 August 2024		Date of mailing of the international search report 19 August 2024
Name and mailing address of the ISA/KR Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon 35208, Republic of Korea Facsimile No. +82-42-481-8578		Authorized officer YANG, Jeong Rok Telephone No. +82-42-481-5709

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2024/027050

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)		Publication date (day/month/year)
US	2011-0270604	A1	03 November 2011	US	8874432 B2	28 October 2014
CN	115687609	A	03 February 2023	None		
CN	115017312	A	06 September 2022	None		
CN	115640808	A	24 January 2023	CN	115640808 B	21 March 2023