DESCRIPTION

TITLE OF INVENTION: TASK PROCESSING METHOD AND TASK PROCESSING SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

The present disclosure claims priority of Chinese Patent Application No. 202310828883.4, filed to China National Intellectual Property Administration on July 06, 2023 and titled "TASK PROCESSING METHOD AND TASK PROCESSING SYSTEM", the content of which is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

The present disclosure relates to the field of computer technologies, and in particular, to a task processing method. At least one embodiment of the present disclosure simultaneously relates to a task processing system, a task processing apparatus, a computing device, a computer-readable storage medium, and a computer program.

BACKGROUND OF THE INVENTION

With the development of computer technologies, establishing an intelligent system that can perform multi-modal conversation interaction like a human being has always been a long-term goal for implementing artificial general intelligence. Although a conventional intelligent system has developed an advanced proficiency in text conversations, the conventional intelligent system still has shortcomings in interactions in more modalities such as voice, pictures, and videos.

Currently, a task inputted by a user is decomposed into several sub-tasks, and a model corresponding to each sub-task is selected from a model library via a redirect, so as to perform sub-task processing through the model corresponding to each sub-task to output a processing result. However, in this solution, the model corresponding to each sub-task is selected from the model library via the redirect results in extremely low task processing efficiency. Therefore, an efficient task processing solution is urgently needed.

SUMMARY OF THE INVENTION

In view of this, the present disclosure provides a task processing method. At least one embodiment of the present disclosure simultaneously relates to a task processing system, a task processing apparatus, a computing device, a computer-readable storage medium, and a computer program, to solve technical defects in the related art.

According to a first aspect of the present disclosure, a task processing method is provided, applied to a service unit in a task processing platform. The task processing platform includes the service unit and multiple processing components. The task processing method includes the following steps.

A task processing request is received, where the task processing request carries data to be processed.

The data to be processed and component description information are inputted into a task processing model to determine at least one target processing component corresponding to the task processing request, where the task processing model is a deep learning model.

The at least one target processing component is called to process the data to be processed, to obtain a task processing result corresponding to the task processing request.

According to a second aspect of the present disclosure, a task processing apparatus is provided, applied to a service unit in a task processing platform. The task processing platform includes the service unit and multiple processing components. The task processing apparatus includes:

a first receiving module, configured to receive a task processing request, where the task processing request carries data to be processed;

a first input module, configured to input the data to be processed and component description information into a task processing model to determine at least one target processing component corresponding to the task processing request, where the task processing model is a deep learning model; and

a first calling module, configured to call the at least one target processing component to process the data to be processed, to obtain a task processing result corresponding to the task processing request.

According to a third aspect of the present disclosure, a task processing system is provided, including a communication interface and a calling unit.

The communication interface is configured to receive a processing request sent by a client, and forward the processing request to the calling unit, and the processing request carries data to be queried.

The calling unit is configured to, in response to the processing request, call a query model to process the data to be queried and processing model description information to determine at least one target processing model corresponding to the data to be queried, where the query model is a deep learning model; and send model information of the at least one target processing model to the client through the communication interface.

According to a fourth aspect of the present disclosure, a computing device is provided, including:

a memory and a processor.

The memory is configured to store computer-executable instructions, the processor is configured to execute the computer-executable instructions, and the computer-executable instructions, when executed by the processor, implement steps of the task processing method provided in the first aspect.

According to a fifth aspect of the present disclosure, a computer-readable storage medium is provided, storing computer-executable instructions. When the computer-executable instructions are executed by a processor, steps of the task processing method provided in the first aspect are implemented.

According to a sixth aspect of the present disclosure, a computer program is provided. When the computer program is executed in a computer, the computer is enabled to perform steps of the task processing method provided in the first aspect.

The task processing method provided in some embodiments of the present disclosure is applied to the service unit in the task processing platform. The task processing platform includes the service unit and the multiple processing components. The task processing request is received, where the task processing request carries data to be processed. The data to be processed and the component description information are inputted into the task processing model to

determine the at least one target processing component corresponding to the task processing request, where the task processing model is the deep learning model. The at least one target processing component is called to process the data to be processed to obtain the task processing result corresponding to the task processing request. The multiple processing components are integrated on the task processing platform, so that the at least one target processing component can be directly called to process the data to be processed, thereby improving task processing efficiency. In addition, the multiple processing components can, on the one hand, enhance a text generation capability of the task processing model in natural language tasks, and on the other hand, compensate for lack of multi-modal generation capabilities of the task processing model, so that a processing capability of the task processing model is more comprehensive.

BRIEF DESCRIPTION OF DRAWINGS

FIG. 1 is a flowchart of a question answering method.

FIG. 2a is an architecture diagram of a first task processing system according to some embodiments of the present disclosure.

FIG. 2b is an architecture diagram of a second task processing system according to some embodiments of the present disclosure.

FIG. 3 is an architecture diagram of a third task processing system according to some embodiments of the present disclosure.

FIG. 4 is a flowchart of a task processing method according to some embodiments of the present disclosure.

FIG. 5 is a schematic diagram of generating sample data in a task processing method according to some embodiments of the present disclosure.

FIG. 6 is a flowchart of an automatic question answering method according to some embodiments of the present disclosure.

FIG. 7 is a flowchart of another task processing method according to some embodiments of the present disclosure.

FIG. 8 is a flowchart of a processing of a task processing method according to some embodiments of the present disclosure.

FIG. 9 is a task example diagram of a task processing method according to some embodiments of the present disclosure.

FIG. 10 is an interface schematic diagram of an automatic question answering interface according to some embodiments of the present disclosure.

FIG. 11 is a structural schematic diagram of a task processing apparatus according to some embodiments of the present disclosure.

FIG. 12 is a structural schematic diagram of an automatic question answering apparatus according to some embodiments of the present disclosure.

FIG. 13 is a structural block diagram of a computing device according to some embodiments of the present disclosure.

DETAILED DESCRIPTION OF THE INVENTION

In the following description, many specific details are described for thorough understanding of the present disclosure. However, the present disclosure can be implemented in many other manners different from those described herein. A person having ordinary skill in the art makes similar improvements without departing from the connotation of the present disclosure. Therefore, the present disclosure is not limited by specific implementations disclosed below.

The terms used in at least one embodiment of the present disclosure are intended to describe the specific embodiments but are not intended to limit at least one embodiment of the present disclosure. The terms “a”, “said”, and “the” in a singular form used in at least one embodiment of the present disclosure and the appended claims are also intended to include a plural form, unless other meanings are clearly indicated in the context. It should be further understood that the term “and/or” used in at least one embodiment of the present disclosure means and includes any or all possible combinations of at least one associated listed item.

It should be understood that although terms “first”, “second”, and the like are used for describing various information in at least one embodiment of the present disclosure, such information should not be limited to these terms. These terms are used for distinguishing between information of the same type. For example, without departing from the scope of the at least one embodiment of the present disclosure, “first” is also referred to as “second”, and

similarly, “second” is also referred to as “first”. Depending on the context, for example, a word “if” used herein is interpreted as “when”, “while”, or “in response to determining”.

In addition, it is to be noted that user information (including but not limited to user equipment information, user personal information, and the like) and data (including but not limited to data for analysis, stored data, displayed data, and the like) involved in at least one embodiment of the present disclosure are all authorized by the user or information and data fully authorized by all parties. The collection, use, and processing of relevant data need to conform to relevant laws, regulations, and standards of relevant countries and regions, and provide corresponding operation portals for the user to choose to authorize or refuse.

In at least one embodiment of the present disclosure, a large model refers to a deep learning model having large-scale model parameters, which usually include hundreds of millions, tens of billions, hundreds of billions, trillions, or even tens of trillions of model parameters. The large model is also referred to as a cornerstone model/foundation model Large-scale pre-training is performed through large-scale unlabeled corpus to produce a pre-trained model with more than 100 million parameters. Such a model can adapt to a wide range of downstream tasks, and the model has good generalization ability, such as a large language model (LLM) and a multi-modal pre-training model.

In a practical application, a small quantity of samples are required to fine-tune the pre-trained model, so that the large model is applied to different tasks. The large model is widely applied to fields such as natural language processing (NLP) and computer vision, and is specifically applied to tasks in the field of computer vision, such as visual question answering (VQA), image caption (IC), and image generation, and tasks in natural language processing fields such as text-based sentiment classification, text abstract generation, and machine translation. Main application scenarios of the large model include a digital assistant, an intelligent robot, search, online education, office software, e-commerce, intelligent design, and the like.

Noun terms in the at least one embodiment of the present disclosure are described first.

Large language model: The large language model refers to a neural network model having billions or even trillions of parameters. These models are used for natural language processing tasks such as language generation, question answering, and text classification. These models

are usually trained through a large amount of text data, and can perform well in multiple different tasks.

Generative artificial intelligence: The generative artificial intelligence (Generative AI) refers to a system that can generate data such as natural language text, an image, an audio, and a video based on an artificial intelligence technology. Different from conventional tasks such as classification and regression, a model in the generative artificial intelligence is creative and imaginative, and generates brand new and unseen content based on an input condition. Common generative AI includes a natural language generation model, an image generation model, and the like.

Tool API integration: The tool API integration refers to an integration manner in which an artificial intelligence model is encapsulated into an application programming interface (API), and is provided for use by a developer or a user. The developer or user uses, through the application programming interface, functions provided by the artificial intelligence model, such as language translation, image recognition, and speech synthesis, to add more abundant functions to an application.

Artificial general intelligence: The artificial general intelligence refers to an artificial intelligence system that has the same broad intelligence capabilities as humans and can continuously learn and adapt to various environments and tasks.

With the development of computer technologies, establishing an intelligent system that can perform multi-modal conversation interaction like a human being has always been a long-term goal for implementing artificial general intelligence. Although a conventional intelligent system has developed an advanced proficiency in text conversations, the conventional intelligent system still has shortcomings in interactions in more modalities such as voice, pictures, and videos.

Referring to FIG. 1, FIG. 1 shows a flowchart of a question answering method. Through the question answering method, descriptions of models in a model library are obtained, and are merged into prompts to establish a connection with a large model, and then the large model serves as a brain to determine an answer to a question of a user. As shown in FIG. 1, a workflow of the question answering method includes four stages: task planning, model selection, task performing, and response generation. At the task planning stage, a user input is decomposed

into several sub-tasks, and a type and difficulty of each sub-task is determined. At the model selection stage, a model library is searched for a suitable model to perform the sub-task. At the task performing stage, the user input is predicted through a selected model. At the response generation stage, a model processing result is collected.

However, in this solution, the model corresponding to each sub-task is selected from the model library via a redirect, and an operation execution process of the model is a tree structure, resulting in extremely low task processing efficiency.

To solve the problem, some embodiments of the present disclosure provide a scalable and learnable large model augmentation framework based on a model-as-a-service sharing platform (Model Scope), which integrates multiple artificial intelligence models as processing components on a task processing platform to form generative artificial intelligence and artificial general intelligence systems, thereby realizing task processing. The model-as-a-service platform provides a developer with a rich, easy-to-use, and low-cost one-stop model service, so that model application is easier. The model service includes abundant pre-trained SOTA models, diverse and open datasets, model reasoning capabilities with one line of code, rapid construction of specialized industry models with ten lines of code, a ready-to-use online development platform, flexible model frameworks and deployment manners, and rich teaching content and technical resources. This enables knowledge question answering, more detailed understanding of requirements, and recommendation of more suitable models, and further provides innovative experiences such as simple tasks, complex tasks, and multiple application programming interface planning (API planning) calls.

Specifically, some embodiments of the present disclosure provide a task processing method applied to a service unit in a task processing platform. A task processing request is received, where the task processing request carries data to be processed. The data to be processed and component description information are inputted into a task processing model to determine at least one target processing component corresponding to the task processing request, where the task processing model is a deep learning model. The at least one target processing component is called to process the data to be processed to obtain a task processing result corresponding to the task processing request. Through application of the solution, the multiple processing components are integrated on the task processing platform, so that the at least one target

processing component is directly called to process the data to be processed, thereby improving task processing efficiency. In addition, the multiple processing components can, on the one hand, enhance a text generation capability of the task processing model in natural language tasks, and on the other hand, compensate for lack of multi-modal generation capabilities of the task processing model, such as video generation, picture generation, a text-to-image conversation, speech synthesis, knowledge question answering, and address structuring, so that a processing capability of the task processing model is more comprehensive.

In the present disclosure, a task processing method is provided. The present disclosure also relates to a task processing system, a task processing apparatus, a computing device, a computer-readable storage medium, and a computer program, which are described in detail one by one in the following embodiments.

Referring to FIG. 2a, FIG. 2a shows an architecture diagram of a first task processing system according to some embodiments of the present disclosure. The task processing system includes a client 100 and a task processing platform 200. The task processing platform 200 includes a service unit 202 and multiple processing components 204.

The client 100 is configured to send a task processing request to the task processing platform 200, where the task processing request carries data to be processed.

The service unit 202 is configured to input the data to be processed and component description information into a task processing model to determine at least one target processing component corresponding to the task processing request, where the task processing model is a deep learning model; and call the at least one target processing component to process the data to be processed to obtain a task processing result corresponding to the task processing request.

In some optional embodiments of the present disclosure, the service unit 200 is further configured to send the task processing result to the client 100. The client 100 is further configured to receive the task processing result sent by the service unit 202.

It is to be noted that the multiple processing components process different types of tasks, such as a video generation task, a picture generation task, a text-to-image conversation task, a speech synthesis task, a knowledge question answering task, an address structuring task, an English-

Chinese translation task, a named entity recognition task, a Chinese-English translation task, and a general information extraction task.

Through application of the solution of the present disclosure, the multiple processing components are integrated on the task processing platform, so that the at least one target processing component is directly called to process the data to be processed, thereby improving task processing efficiency. In addition, the multiple processing components can, on the one hand, enhance a text generation capability of the task processing model in natural language tasks, and on the other hand, compensate for lack of multi-modal generation capabilities of the task processing model, so that a processing capability of the task processing model is more comprehensive.

Referring to FIG. 2b, FIG. 2b shows an architecture diagram of a second task processing system according to some embodiments of the present disclosure. The task processing system includes a communication interface 206 and a calling unit 208.

The communication interface 206 is configured to receive a processing request sent by a client, and forward the processing request to the calling unit 208, where the processing request carries data to be queried.

The calling unit 208 is configured to, in response to the processing request, call a query model to process the data to be queried and processing model description information to determine at least one target processing model corresponding to the data to be queried, where the query model is a deep learning model; and send model information of the at least one target processing model to the client through the communication interface 206.

Specifically, the model information is a model file of the target processing model. The client locally constructs the target processing model based on the model file of the target processing model. The model information is further model call notification information of the target processing model. The client calls the target processing model through a model call interface in response to the model call notification information. The model information is further model introduction information of the target processing model. The model introduction information includes, but is not limited to, model description information, a model use manner, a model use range, a model running environment, a model code example, and a model training data introduction.

The data to be queried includes model query requirement information. A query model is a deep learning model. The query model is trained based on sample data, or a pre-trained large model (such as a large language model) is set as the query model. A processing model is a model having different functions, such as a video generation model, a picture generation model, a text-to-image conversation model, a speech synthesis model, a knowledge question answering model, an address structuring model, an English-Chinese translation model, a named entity recognition model, a Chinese-English translation model, and a general information extraction model. The processing model description information is used for describing the processing model, including but not limited to, a model name, a model function, and a model type, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this.

In some optional embodiments of the present disclosure, after the calling unit sends the model information of the at least one target processing model to the client through the communication interface, a user directly performs task processing based on the model information. Further, the user further sends model adjustment parameters through the client based on an actual requirement. The calling unit performs secondary development and adjustment on the at least one target processing component based on the model adjustment parameters. In other words, the calling unit is further configured to adjust the at least one target processing model based on the model adjustment parameters, and send the model information of the at least one adjusted target processing model to the client through the communication interface. The model adjustment parameters include, but is not limited to, a model adjustment sample set, a model adjustment weight, and the like. The adjustment sample set includes multiple adjusted samples, and each adjusted sample carries an adjusted sample label.

Referring to FIG. 3, FIG. 3 shows an architecture diagram of a third task processing system according to some embodiments of the present disclosure. The task processing system includes multiple clients 100 and a task processing platform 200. A communication connection is established between the multiple clients 100 through the task processing platform 200. In a task processing scenario, the task processing platform 200 is configured to provide a task processing service between the multiple clients 100. The multiple clients 100 are respectively used as sending terminals or receiving terminals, to implement communication through the task processing platform 200.

A user interacts with the task processing platform 200 through the client 100 to receive data sent by another client 100 or to send data to the other client 100, or the like. In the task processing scenario, the user publishes a data stream to the task processing platform 200 through the client 100. The task processing platform 200 generates a task processing result based on the data stream, and pushes the task processing result to another client with which communication is established. A connection is established between the client 100 and the task processing platform 200 through a network. The network provides a medium for a communication link between the client 100 and the task processing platform 200. The network includes various connection types, for example, a wired communication link, a wireless communication link, or an optical cable. Data transmitted by the client 100 needs to be encoded, transcoded, compressed, and the like before being published to the task processing platform 200.

The client 100 is a browser, an application (APP), or a web application such as a hypertext markup language 5 (H5) application, a light application (also referred to as a mini program, a lightweight application), or a cloud application. The client 100 is developed and obtained based on a software development kit (SDK) of a corresponding service provided by the task processing platform 200, such as a real-time communication (RTC)-based SDK. The client 100 is deployed in an electronic device, and needs to run depending on a device, some APPs in the device, or the like. For example, the electronic device has a display screen and supports information browsing, which is a personal mobile terminal such as a mobile phone, a tablet computer, or a personal computer. Various other types of applications, such as a man-machine conversation application, a model training application, a text processing application, a web browser application, a shopping application, a search application, an instant messaging tool, an email client, and social platform software, is usually configured in the electronic device.

The task processing platform 200 includes servers that provide various services, for example, a server that provides communication services for multiple clients, for another example, a server for background training that provides support for a model used on a client, and for another example, a server that processes data sent by a client. It is to be noted that the task processing platform 200 is implemented as a distributed server cluster formed by multiple servers, or is implemented as a single server. The server is also a server of a distributed system, or a server combined with a blockchain. The server is also a cloud server of basic cloud

computing services such as a cloud service, a cloud database, cloud computing, a cloud function, cloud storage, a network service, cloud communication, a middleware service, a domain name service, a security service, a content delivery network (CDN), and a big data and artificial intelligence platform, or is an intelligent cloud computing server or an intelligent cloud host with an artificial intelligence technology.

It is worth mentioning that the task processing method provided in the present disclosure is generally performed by the task processing platform. However, in some other embodiments of the present disclosure, the client also has a similar function to the task processing platform, thereby performing the task processing method provided in the present disclosure. In some other embodiments, the task processing method provided in the present disclosure is further jointly performed by the client and the task processing platform.

Referring to FIG. 4, FIG. 4 is a flowchart showing a task processing method according to some embodiments of the present disclosure. The task processing method is applied to a service unit in a task processing platform. The task processing platform includes the service unit and multiple processing components. The task processing method specifically includes the following steps.

In step 402: a task processing request is received, where the task processing request carries data to be processed.

In at least one embodiment of the present disclosure, the data to be processed is obtained, and then task processing is performed based on the data to be processed to obtain a task processing result.

Specifically, the data to be processed is data in different scenarios, such as data to be processed in a conference scenario and data to be processed in an e-commerce scenario. Tasks corresponding to the data to be processed are different types of tasks, such as a video generation task, a picture generation task, a text-to-image conversation task, a speech synthesis task, a knowledge question answering task, an address structuring task, an English-Chinese translation task, a named entity recognition task, a Chinese-English translation task, and a general information extraction task. The data to be processed includes, but is not limited to, data content, a task execution time, a task identifier, and the like, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this.

In a practical application, the task processing request is received in multiple manners, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this. In a possible implementation of the present disclosure, the task processing request sent by the user through the client is received. In another possible implementation of the present disclosure, the task processing request regularly sent by the client is received.

In step 404: the data to be processed and component description information are inputted into a task processing model to determine the at least one target processing component corresponding to the task processing request, where the task processing model is a deep learning model.

In at least one embodiment of the present disclosure, after the task processing request is received, the data to be processed and the component description information are inputted into the task processing model to determine the at least one target processing component corresponding to the task processing request, where the task processing model is the deep learning model.

Specifically, the task processing model is the deep learning model. The task processing model is trained based on sample data, or a pre-trained large model (such as a large language model) is used as the task processing model. The multiple processing components in the task processing platform are integrated into the task processing platform by encapsulating an artificial intelligence model into an API interface in a manner of Tool API integration, and are further pluggable components pre-configured on the task processing platform by a configuration party. The processing components are components having different functions, such as a component supporting translation, a component supporting video generation, and a component supporting picture generation. The component description information is used for describing the processing component, including but not limited to, a component name, a component function, and a component type, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this.

It is to be noted that multiple manners are provided for inputting the data to be processed and the component description information into the task processing model to determine the at least one target processing component corresponding to the task processing request, which is

specifically selected based on an actual situation. The present disclosure does not impose any limitation on this.

In a possible implementation of the present disclosure, the data to be processed and the component description information of each of the processing components in the task processing platform are directly inputted into the task processing model, to determine the at least one target processing component corresponding to the task processing request.

In another possible implementation of the present disclosure, to reduce the processing pressure of the task processing model, the multiple processing components in the task processing platform are screened to determine at least one candidate processing component. The data to be processed and the component description information of the at least one candidate processing component are inputted into the task processing model to determine the at least one target processing component corresponding to the task processing request. The multiple processing components are screened in multiple manners. For example, through comparison of a task type of the task processing request with a component type of each processing component, the at least one candidate processing component matching the task type is screened out from the multiple processing components. Alternatively, through comparison of the task description information of the task processing request with the component function information of each processing component, the at least one candidate processing component matching the task description information is screened out from the multiple processing components.

Further, the data to be processed and the component description information are inputted into the task processing model. The task processing model determines the at least one target processing component corresponding to the task processing request through a semantic understanding capability.

In some optional embodiments of the present disclosure, coarse ranking is performed on the data to be processed and local knowledge to quickly determine the task processing result. In the case of a failure of the coarse ranking, the data to be processed and the component description data of the processing component are further finely ranked. In other words, before the data to be processed and the component description information are inputted into the task

processing model to determine the at least one target processing component corresponding to the task processing request, the following steps are further included.

Reference text data is obtained.

The data to be processed is matched with the reference text data to obtain a matching result.

An operation of inputting the data to be processed and the component description information into the task processing model to determine the at least one target processing component corresponding to the task processing request includes the following step.

In response to the matching result indicating a matching failure, the data to be processed and the component description information are inputted into the task processing model to determine the at least one target processing component corresponding to the task processing request.

Specifically, the reference text data is local knowledge. The reference text data includes, but is not limited to, a task processing prompt text, a reply template, and the like. The reference text data is obtained in multiple manners. The reference text data inputted by a user is received, or the reference text data is read from another data obtaining device or database. The reference text data is updated in real time.

In a practical application, multiple manners are provided for matching the data to be processed with the reference text data to obtain a matching result, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this. In a possible implementation of the present disclosure, sparse retrieval is performed. In other words, a similarity between the data to be processed and the reference text data is calculated in a word dimension. In another possible implementation of the present disclosure, vector retrieval is performed. In other words, context semantic information is incorporated to calculate a similarity between the data to be processed and the reference text data in a statement dimension. In yet another possible implementation of the present disclosure, the data to be processed and the reference text data are inputted into the task processing model to obtain the matching result.

It is to be noted that the data to be processed is matched with the reference text data to obtain the matching result. In response to the matching result indicating the matching failure, it

indicates that the reference text data does not include data related to the data to be processed. In other words, the task processing result corresponding to the task processing request cannot be quickly determined through the reference text data, and at least one target processing component corresponding to the task processing request needs to be determined through the task processing model, so as to process the data to be processed through the target processing component. In response to the matching result indicating the successful match, it indicates that the reference text data includes the data related to the data to be processed. In other words, the task processing result corresponding to the task processing request can be quickly determined through the reference text data without the need to call the processing component.

Through application of the solution of the present disclosure, the reference text data is obtained. The data to be processed is matched with the reference text data to obtain the matching result. In response to the matching result being a matching failure, the data to be processed and component description information are inputted into the task processing model to determine the at least one target processing component corresponding to the task processing request. Scalable knowledge retrieval and component selection are proposed. Through coarse ranking based on retrieval and fine ranking based on the processing component, extended local knowledge and an extended processing component are incorporated into the output of the task processing model, thereby improving scalability of task processing. In addition, since the processing component and the local knowledge are updated, extended, and enhanced in real time, the task processing process can be updated, thereby further improving real-time performance, adaptability, and compatibility of task processing.

Further, after the data to be processed is matched with the reference text data to obtain the matching result, the following step is further included.

In response to the matching result indicating a matching success, the data to be processed and the reference text data are inputted into the task processing model to determine the task processing result corresponding to the task processing request.

It is to be noted that when the matching result is the successful match, the task processing result corresponding to the task processing request is determined based on the data to be processed and the reference text data.

In a practical application, the task processing result corresponding to the task processing request is determined based on the data to be processed and the reference text data in multiple manners, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this.

In a possible implementation of the present disclosure, first, multiple indexes are created for the reference text data. One index corresponds to multiple reference texts. After the multiple indexes are created, a retriever is further created based on the multiple indexes. A type of the retriever includes at least one of sparse retrieval and vector retrieval. Then the data to be processed is inputted into the retriever. The retriever returns retrieved knowledge related to the data to be processed. Finally, the retrieved knowledge returned by the retriever is used as the task processing result.

The retriever calculates a similarity between the data to be processed and the reference text data when performing the sparse retrieval and the vector retrieval, thereby determining the task processing result corresponding to the task processing request based on the similarity. A manner of calculating the similarity between the data to be processed and the reference text data includes, but is not limited to, a cosine similarity, term frequency-inverse document frequency (TF-IDF) calculation, and levenshtein distance calculation.

In another possible implementation of the present disclosure, the data to be processed and the reference text are inputted into the task processing model to determine the task processing result corresponding to the task processing request.

Through application of the solution of the present disclosure, when the matching result is a successful match, the data to be processed and the reference text data are inputted into the task processing model to determine the task processing result corresponding to the task processing request, so that the task processing result is quickly determined, thereby improving task processing efficiency.

In some optional embodiments of the present disclosure, a training manner of the task processing model include the following steps.

A sample set is obtained, where the sample set includes the multiple samples, each sample carries a sample label, and each sample is constructed based on at least one of sample text data, sample question-answer pair data, and sample component call data.

A first sample is extracted from the sample set, where the first sample is any one of the multiple samples.

The first sample is inputted into the task processing model to obtain a first prediction result of the first sample.

A first loss value is calculated based on the first prediction result and a first sample label carried in the first sample.

Model parameters of the task processing model are adjusted based on the first loss value, and an operation of extracting the first sample from the sample set is returned to be executed, until a first preset stopping condition is reached to obtain the task processing model resulting from the training method.

Specifically, the training manner of the task processing model is supervised training. The task processing model is trained based on the multiple samples and the sample label carried in each sample data. The sample label is a true label of each sample, or is understood as a processing target of the task processing model, and is used for guiding a process of training the task processing model.

In a possible implementation of the present disclosure, the first preset stop condition includes that the first loss value is less than or equal to a first preset threshold. After the first loss value is calculated based on the first prediction result and the first sample label carried in the first sample, the first loss value is compared with the first preset threshold.

Specifically, in response to the first loss value being greater than the first preset threshold, it indicates that a relatively large difference exists between the first prediction result and the first sample label, and the task processing model has relatively poor prediction capability for the first prediction result. In this case, parameters of the task processing model are adjusted, and the step of extracting the first sample from the sample set is performed to continually train the task processing model until the first loss value is less than or equal to the first preset threshold, which indicates that a relatively difference exists between the first prediction result and the

first sample label, and the task processing model has relatively strong prediction capability. When the first preset stop condition is satisfied, the training is stopped to obtain a trained task processing model.

In another possible implementation of the present disclosure, in addition to the comparison of a magnitude relationship between the first loss value and the first preset threshold, it is further determined based on a quantity of first iterations whether training of a current task processing model is completed.

Specifically, in response to the first loss value being greater than the first preset threshold, the parameters of the task processing model are adjusted, the step of extracting the first sample from the sample set is performed to continually train the task processing model. Until a first preset quantity of iterations is reached, the iterations are stopped to obtain the trained task processing model. The first preset threshold and the first preset quantity of iterations are specifically selected based on an actual situation. The present disclosure does not impose any limitation on this.

In a practical application, the first loss value is calculated through many functions, such as a cross-entropy loss function, an L1 norm loss function, a maximum loss function, a mean square error loss function, and a logarithm loss function, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this.

Through application of the solution of the present disclosure, the first loss value is calculated based on the first prediction result and the first sample label carried in the first sample. The first loss value is compared with the first preset stop condition. The task processing model continues to be trained when the first preset stop condition is not satisfied, until the first preset stop condition is satisfied, to complete training to obtain the task processing model. The parameters of the task processing model are continuously adjusted, so that a finally obtained task processing model is more accurate.

It is to be noted that the sample set is obtained in multiple manners, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this. A large amount of samples carrying sample labels are read from another data obtaining device or database to form a training set. A large amount of samples carrying sample labels that are inputted by a user are further received to form a sample set.

In a practical application, high-quality annotated data (sample data) is a key factor in determining an upper limit of model performance. As a quantity of model parameters increases, a requirement for the annotated data also increases. However, a process of obtaining sample data usually is time-consuming and labor-intensive, and incurs considerable costs. This seriously affects iteration efficiency of a model. Therefore, in the present disclosure, each sample is constructed based on at least one of the sample text data, the sample question-answer pair data, and the sample component call data.

Specifically, the sample text data, the sample question-answer pair data, and the sample component call data are obtained. A seed file is constructed based on the sample text data, the sample question-answer pair data, and the sample component call data. Then the seed file is processed through a data processing strategy. More sample data is generated through great generalization ability of the task processing model for fine tuning of the task processing model, thereby greatly improving scale and precision of data annotation. The data processing strategy includes, but is not limited to, at least one of zero-shot learning, few-shot learning, and cot prompt engineering.

Referring to FIG. 5, FIG. 5 shows a schematic diagram of generating sample data in a task processing method according to some embodiments of the present disclosure. As shown in FIG. 5, sample text data, sample question-answer pair data, and sample component call data are obtained. The sample question-answer pair data is obtained through zero-shot learning and few-shot learning. The sample question-answer pair data is fused to construct sample conversation data, and the sample component call data is further processed to construct the sample conversation data. The sample conversation data is used as the sample data.

In a practical application, when calculating the first loss value based on the first prediction result and the first sample label carried in the first sample, the first loss value is calculated directly based on the first prediction result and the first sample label. The first prediction result is further parsed based on model knowledge distillation to calculate the first loss value through a weight. In other words, the calculation of the first loss value based on the first prediction result and the first sample label carried in the first sample includes the following steps.

The first prediction result is parsed to obtain multiple first prediction sub-results.

A weight corresponding to each of the first prediction sub-results is determined based on a type of each of the first prediction sub-results.

The first loss value is calculated based on the multiple first prediction sub-results, the weight corresponding to each of the first prediction sub-results, and the first sample label carried in the first sample.

It is to be noted that multiple manners are provided for parsing the first prediction result to obtain the multiple first prediction sub-results, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this. In a possible implementation of the present disclosure, the first prediction result is matched with a preset parsing dictionary, thereby splitting the first prediction result to obtain multiple first prediction sub-results. In another possible implementation of the present disclosure, the first prediction result is parsed through a preset word segmentation tool to obtain multiple first prediction sub-results.

Further, after the multiple first prediction sub-results are obtained, the weight corresponding to each of the first prediction sub-results is set. During setting of the weight, the setting is performed based on a type of each first prediction sub-result, and the weight corresponding to each first prediction sub-result is further determined based on an actual requirement. Finally, the first loss value is calculated through weighting based on the multiple first prediction sub-results, the weight corresponding to each first prediction sub-result, and the first sample label carried in the first sample.

For example, problems such as generation of a component name and calling correctness exist during calling of a processing component, and during multiple rounds of task processing, incoming component call parameters need to be searched during previous several rounds of task processing, resulting in a problem of a parameter transfer error. Therefore, the first prediction result is parsed to obtain the multiple first prediction sub-results, such as a component name, a component call parameter, component content, and component reply content in a component reply. Weights are respectively set for the first prediction sub-results, and the loss value is calculated through weighting.

Through application of the solution of the present disclosure, the first prediction result is parsed to obtain the multiple first prediction sub-results. The weight corresponding to each of the first

prediction sub-results is determined based on the type of each first prediction sub-result. The first loss value is calculated based on the multiple first prediction sub-results, the weight corresponding to each first prediction sub-result, and the first sample label carried in the first sample, so as to adjust a proportion of the first prediction sub-result in a model training process based on the weight. On the one hand, a task completion rate can be increased, and on the other hand, correctness of parameter transfer can be improved.

In step 406: the at least one target processing component is called to process the data to be processed, to obtain a task processing result corresponding to the task processing request.

In at least one embodiment of the present disclosure, after the task processing request is received, and the data to be processed and the component description information are inputted into the task processing model to determine the at least one target processing component corresponding to the task processing request. Further, the at least one target processing component is called to process the data to be processed to obtain the task processing result corresponding to the task processing request.

Through application of the solution of the present disclosure, the multiple processing components are integrated on the task processing platform, so that the at least one target processing component is directly called to process the data to be processed, thereby improving task processing efficiency. In addition, the multiple processing components can, on the one hand, enhance a text generation capability of the task processing model in natural language tasks, and on the other hand, compensate for lack of multi-modal generation capabilities of the task processing model, so that a processing capability of the task processing model is more comprehensive.

In some optional embodiments of the present disclosure, since the data to be processed does not meet a parameter input requirement of the target processing component, format conversion is performed on the data to be processed based on the component call parameters of the target processing component. In other words, before calling the at least one target processing component to process the data to be processed to obtain the task processing result corresponding to the task processing request, the task processing method further includes the following steps.

Component call parameters of the at least one target processing component are obtained.

Format conversion is performed on the data to be processed based on the component call parameters to obtain updated data to be processed.

An operation of calling the at least one target processing component to process the data to be processed, to obtain the task processing result corresponding to the task processing request includes the following step.

The at least one target processing component is called to process the updated data to be processed, to obtain the task processing result corresponding to the task processing request.

Specifically, the component call parameters are used for describing a calling rule of the target processing component. The component call parameters includes, but is not limited to, a language type, a data format, and the like supported by the target processing component. The component call parameters of the at least one target processing component are obtained in multiple manners. For one example, component call parameters of the at least one target processing component sent by a configuration party are received. For another example, component call parameters of the at least one target processing component are read from another data obtaining device or database.

It is to be noted that after the component call parameters of the at least one target processing component are obtained, further, it is determined whether the data to be processed is consistent with the component call parameters. In response to the data to be processed being consistent with the component call parameters, it indicates that the target processing component directly processes the data to be processed without the need to perform format conversion on the data to be processed, and the at least one target processing component is called to process the data to be processed to obtain the task processing result corresponding to the task processing request. In response to the data to be processed not being consistent with the component call parameters, it indicates that the target processing component cannot directly process the data to be processed. Format conversion is performed on the data to be processed based on the component call parameters to obtain the updated data to be processed. The at least one target processing component is called to process the updated data to be processed to obtain the task processing result corresponding to the task processing request.

In a practical application, multiple manners are provided for performing the format conversion on the data to be processed based on the component call parameters to obtain the updated data

to be processed, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this. In a possible implementation of the present disclosure, the data to be processed is converted through a format conversion tool. For example, Chinese data to be processed is translated into an English form through a translation tool. In another possible implementation of the present disclosure, multiple processing components are searched for a conversion processing component that performs format conversion on the data to be processed to enable the data to be processed to be consistent with the component call parameters. The format conversion is performed on the data to be processed based on the component call parameters through a conversion processing component, to obtain updated data to be processed. For example, the Chinese data to be processed is translated into the English form through the translation model.

Through application of the solution of the present disclosure, the component call parameters of the at least one target processing component are obtained. The format conversion is performed on the data to be processed based on the component call parameters to obtain the updated data to be processed. The at least one target processing component is called to process the updated data to be processed to obtain the task processing result corresponding to the task processing request. It is ensured that the at least one target processing component is successfully called, thereby ensuring a success rate of task processing.

In some optional embodiments of the present disclosure, the task processing model is optimized through the call information of the at least one target processing component, to implement reinforcement learning based on feedback from the processing component. In other words, after calling the at least one target processing component to process the data to be processed to obtain the task processing result corresponding to the task processing request, the task processing method further includes the following step.

Call information of the at least one target processing component is obtained, and the task processing model is optimized based on the call information.

Specifically, the call information of the target processing component includes, but is not limited to, a calling process condition, a calling result, and a task processing result.

It is to be noted that in the process of calling the at least one target processing component to process the data to be processed, the at least one target processing component is monitored to

obtain the call information of the at least one target processing component. Further, the call information is used as a reward signal. The task processing model is optimized through an actor-critic algorithm, thereby optimizing a decision-making process.

Through application of the solution of the present disclosure, the call information of the at least one target processing component is obtained, and the task processing model is optimized based on the call information, so that the task processing model determines the at least one target processing component corresponding to the task processing request more accurately, and further accurately determines a processing order of the at least one target processing component, thereby accurately calling the at least one target processing component to implement task processing, and improving the generalization ability of the task processing model.

In some optional embodiments of the present disclosure, the task processing request further carries task requirement information. After calling the at least one target processing component to process the data to be processed to obtain the task processing result corresponding to the task processing request, the task processing method further includes the following step.

The task requirement information and the task processing result are inputted into the task processing model to obtain an updated task processing result, where the updated task processing result is consistent with the task requirement information.

Specifically, the task requirement information is used for defining the task processing result based on an actual task requirement. The task requirement information includes, but is not limited to, a word count requirement of the task processing result and a language requirement of the task processing result, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this.

For example, it is assumed that the task processing result is in English, and a result of the task requirement information is in Chinese, the task requirement information and the task processing result are inputted into the task processing model, and the task processing model translates the English task processing result into a Chinese task processing result. It is assumed that the task processing result has excessively many words, the task requirement information and the task processing result are inputted into the task processing model, and the task

processing model extracts main idea of the task processing result to obtain an updated task processing result.

Through application of the solution of the present disclosure, the task requirement information and the task processing result are inputted into the task processing model to obtain the updated task processing result, to ensure that the task processing result is consistent with the task requirement information, thereby improving accuracy of task processing and user experience.

In some optional embodiments of the present disclosure, after calling the at least one target processing component to process the data to be processed to obtain the task processing result corresponding to the task processing request, the task processing method further includes the following step.

The task processing result is sent to a client, so that the client displays the task processing result to a user.

It is to be noted that after calling the at least one target processing component to process the data to be processed to obtain the task processing result corresponding to the task processing request, the task processing result is sent to the client, and a processing manner corresponding to the task processing result is further sent to the client. The processing manner is processing based on the reference text data or processing by calling the target processing component. In response to the processing manner being processing by calling the target processing component, the processing manner further includes component information of the target processing component, such as a quantity of components, a component source, and a component source address. Further, in response to the task processing result being the updated task processing result, the task processing result and the updated task processing result are sent to the client simultaneously.

In a practical application, the client displays the task processing result to the user in multiple manners, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this. In a possible implementation of the present disclosure, the client displays the task processing result to the user. In another possible implementation of the present disclosure, the client displays the task processing result and the processing manner to the user simultaneously, so that the user can accurately know the task processing result and a processing manner corresponding to this task processing result. In yet another possible

implementation of the present disclosure, the task processing result and the updated task processing result are displayed to the user simultaneously.

It is worth mentioning that after the task processing result corresponding to the task processing request is obtained, quality inspection is performed on the task processing result. The task processing result is sent to the client when the quality inspection passes, and task processing is performed again when the quality inspection fails.

Through application of the solution of the present disclosure, the task processing result is sent to the client, to enable the client to display the task processing result to the user, so that the user accurately obtains the task processing result, thereby increasing interaction with the user, and enhancing user satisfaction.

In some optional embodiments of the present disclosure, the task processing request further carries display requirement information. An operation of sending the task processing result to the client includes the following steps.

The task processing result is sent to the client based on the display requirement information.

Specifically, the display requirement information is used for describing a requirement of a user for viewing the task processing result. The display requirement information includes, but is not limited to, display of the task processing result, display of the task processing result and the processing manner corresponding to this task processing result, display of the task processing result obtained through processing by the target processing component and the component information of the target processing component, display of the task processing result obtained through processing based on the reference text data, or the like, which is specifically set based on an actual requirement of the user. The present disclosure does not impose any limitation on this.

It is to be noted that when the processing manner corresponding to the data to be processed includes both the processing based on the reference text data and the processing through the target processing component, a first task processing result is generated based on the reference text data, and a second task processing result is generated through the target processing component. Next, the first task processing result and the second task processing result are displayed to the user. Further, after the first task processing result and the second task

processing result are generated, display options of “display of a result obtained by processing through the target processing component” and “display of a result obtained through processing based on the reference text data” are further provided to the user, a target display option selected by the user is received, and the task processing result corresponding to the target display option is displayed.

Through application of the solution of the present disclosure, the task processing result is sent to the client based on the display requirement information, so that the client displays the task processing result to the user, thereby increasing interaction with the user, and enhancing user satisfaction.

In some optional embodiments of the present disclosure, after the client displays the task processing result to the user, the user performs data processing based on the task processing result, or sends a post-processing request based on the task processing result displayed by the client to perform multiple rounds of task processing. In other words, after sending the task processing result to the client, the task processing method further includes the following steps.

A post-processing request sent by the user is received based on the task processing result. The post-processing request is processed to obtain a post-processing result corresponding to the post-processing request.

Specifically, the post-processing request refers to the task processing request sent by the user based on the task processing result after obtaining the task processing result. The data requested for processing by the post-processing request is new data to be processed, and is further data to be processed carried in the task processing request, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this.

It is to be noted that when the data requested for processing by the post-processing request is the data to be processed carried in the task processing request, it indicates that the user is dissatisfied with the task processing result, and the data to be processed needs to be processed again. Further, the post-processing request sent by the user based on the task processing result carries a processing manner specified by the user. For example, in response to a task display result being generated based on the reference text data, the post-processing request sent by the user based on the task processing result requests processing of the data to be processed through the target processing component. In response to the task display result being generated by the

target processing component, the post-processing request sent by the user based on the task processing result requests processing of the data to be processed based on the reference text data. In a practical application, an implementation of "processing the post-processing request to obtain a post-processing result corresponding to the post-processing request" is the same as an implementation of the task processing method, and details are not described in the present disclosure again.

Through application of the solution of the present disclosure, the post-processing request sent by the user based on the task processing result is received, and the post-processing request is processed to obtain the post-processing result corresponding to the post-processing request, thereby increasing interaction with the user, and enhancing user satisfaction.

Application of the task processing method provided in the present disclosure in the field of intelligent question answering is used as an example below with reference to FIG. 6, to further describe the task processing method. FIG. 6 is a flowchart of an automatic question answering method according to some embodiments of the present disclosure. The automatic question answering method is applied to a service unit in a task processing platform. The task processing platform includes the service unit and multiple processing components. The task processing method specifically includes the following steps.

In step 602: a question to be answered is obtained.

In step 604: the question to be answered and component description information are inputted into a task processing model, to determine at least one target processing component corresponding to the question to be answered, where the task processing model is a deep learning model for natural language.

In step 606: the at least one target processing component is called to process the question to be answered, to obtain a reply result corresponding to the question to be answered, and the reply result is sent to a user.

In step 608: an updated question to be answered sent by the user is received based on the reply result, and the step of inputting the question to be answered and the component description information into the task processing model to determine the at least one target processing

component corresponding to the question to be answered is performed, until a second preset stop condition is satisfied, to obtain the reply result.

It is to be noted that the question to be answered is data to be processed, and the reply result is a task processing result corresponding to a task processing request. The implementations of step 602 to step 606 are the same as the implementations of step 402 to step 406, and details are not described in the present disclosure again.

In a practical application, the automatic question answering process includes a single round of question answering, or includes rounds of question answering. The question to be answered is a general information retrieval question, a picture generation question, and is further a text-to-video generation question. The question to be answered is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this. The second preset stop condition includes that the user stops sending the updated question to be answered, and a second preset quantity of iterations is reached. The second preset stop condition and the second preset quantity of iterations are specifically selected based on an actual situation. The present disclosure does not impose any limitation on this.

For example, it is assumed that the automatic question answering process is four rounds of question answering. Question 1: Write a Chinese advertising slogan for **A** commodity. Answer 1: **A** 商品, 让你的生活更加精彩. Question 2 (first question update): Help me translate this answer into English. Answer 2: **A** commodity, make your life more colorful. Question 3 (second question update): Based on the advertising slogan, help me generate a picture. Answer 3: OK, I will call a picture generation component to draw a picture for you (the picture displayed). Question 4 (third question update): Based on the advertising slogan, help me generate a video. Answer 4: The task processing platform has a video generation component supporting English. Therefore, I will first translate your question into English through a Chinese-English translation model and then call an English video generation model, to obtain a translated advertising slogan is **A** commodity, make your life more colorful, and a video is being generated through the video generation model (the video displayed).

It is assumed that the automatic question answering is two rounds of knowledge question answering. Question 1: Is there a model that supports an open-domain conversation? Answer 1: Yes, we have a model named X for the open-domain conversation. The model receives a

user conversation history and external knowledge related to the conversation, user persona, and model persona information, and output an appropriate text reply. You can use the model through a code example provided by us. Question 2: How to use the model? Answer 2: First, you need to install the latest version of our application, and then you can use the model X. You can use the model through the code example provided by us.

It is worth mentioning that the automatic question answering method is applied to a chatbot. The chatbot has an intelligent operation solution, and provides a user with a human-like conversation experience. The chatbot can start a conversation simply once connected to knowledge without the need for knowledge processing.

Specifically, once the knowledge is connected, a conversation is started without knowledge processing. A type of knowledge supported by the chatbot includes, but is not limited to, a document, a data table, and a project process. The document knowledge is obtained through a direct connection to an enterprise-level website, an enterprise knowledge base, or the like, and is further obtained through importing of a document. The chatbot redefines frequent asked questions, and provide intelligent operation for the user.

A human-like conversation experience: It means that language comprehension ability of the chatbot is greatly improved. The chatbot implements joint understanding of different types of knowledge, and has multi-round conversation and reasoning analysis capabilities. The conversation process of the chatbot is more natural and human-like, and the generated conversation is more coherent and natural. In the knowledge question answering process, an original knowledge fragment is cited as a reference.

Intelligent operation solution: The chatbot has a complete guide for beginners and glossary, providing a one-stop introduction to usage, so as to achieve step-by-step guidance. The chatbot supports annotation of replies of the robot and support the use of high-frequency documents to ensure an effect, so that the effect can be intervened. The chatbot further provides data such as brief and practical session overview data and project analysis through a data report, thereby providing feedback for the effect.

Through application of the solution of the present disclosure, the multiple processing components are integrated on the task processing platform, so that the at least one target processing component is directly called to process the question to be answered, thereby

improving automatic question answering efficiency. In addition, the multiple processing components can, on the one hand, enhance a text generation capability of the task processing model in natural language tasks, and on the other hand, compensate for lack of multi-modal generation capabilities of the task processing model, so that the automatic question answering is more comprehensive.

Referring to FIG. 7, FIG. 7 is a flowchart of another task processing method according to some embodiments of the present disclosure. As shown in FIG. 7, a solution provided in the present disclosure includes three parts, which are respectively a front end, a task processing platform, and a model-as-a-service sharing platform. The front end is configured to provide a task processing window for a user to communicate with the task processing platform. The task processing platform is configured to provide reference text data, a component service, a component list, and component description information that are provided by the model-as-a-service sharing platform for the user to use through a conversation. The model-as-a-service sharing platform is configured to provide a processing component service for calling by a learnable large model augmentation framework, and provides reference text data for generating a task processing result.

A processing of the task processing platform is divided into knowledge search and component selection. Reference text data and a target processing component corresponding to data to be processed inputted by a user are found. Specifically, through the knowledge search, a knowledge fragment to be used is selected from the reference text based on the data to be processed inputted by the user. Through the component selection, the target processing component to be used is selected from multiple processing components based on the data to be processed inputted by the user.

The learnable large model augmentation framework completes the task processing request based on the data to be processed, the reference text data, and the multiple processing components, and provides the task processing result. Specifically, the learnable large model augmentation framework is divided into three parts. A first part is task planning, which is to analyze a user request through a task processing model to understand the intent of the user, and determine the target processing component to be used and a component call parameter. A second part is task execution, which is to call and execute the target processing component,

and return an execution result outputted by the target processing component to the task processing model. A third part is answer generation, which is to generate a processing result based on the execution result, and generate new task planning until the task is completed.

Through application of the solution of the present disclosure, a large generative model based on a processing component and retrieval augmentation is provided, and domain knowledge and an updated processing component are obtained in real time through knowledge search and component selection, achieving real-time performance and updateability. Based on the learnable large model augmentation framework, the generalization ability of the task processing model is improved through sample data construction, model knowledge distillation, and reinforcement learning based on component feedback. In addition, task processing efficiency is improved through a task processing flow of a sequence result, thereby implementing knowledge question answering, understanding of more detailed requirements, and model recommendation. Further, a simple task, a complex task, and multi-application programming interface planning and calling are implemented, and a developer further integrates, through the task processing flow, a processing component developed by the developer and provides the processing component to the user for use.

Referring to FIG. 8, FIG. 8 is a flowchart of a processing process of a task processing method according to some embodiments of the present disclosure. A task processing request is received, where the task processing request carries data to be processed. The data to be processed is inputted into a task processing model to determine whether to perform task processing through component selection or to perform task processing through knowledge search.

In response to the task processing being performed through the component selection, the data to be processed and component description information are inputted into the task processing model to determine at least one target processing component corresponding to the task processing request and component call parameters of a target processing component. The at least one target processing component is called, data to be processed that is consistent with the component call parameters is inputted into the at least one target processing component, and an execution result outputted by the target processing component is inputted into the task processing model, to generate a task processing result corresponding to the task processing

request. The target processing component obtains pre-stored local knowledge from a database and processes the data to be processed based on the local knowledge. The local knowledge includes, but is not limited to, a picture, an audio, and a video.

In response to the task processing being performed through the knowledge search, the data to be processed is encoded to generate a processing vector corresponding to the data to be processed. A reference vector is obtained from a vector repository. A similarity between the processing vector and the reference vector is calculated. A prompt text is further generated based on the similarity. The prompt text is inputted into the task processing model to obtain a task processing result corresponding to the data to be processed. The reference vector is obtained through encoding of reference text data in the local knowledge.

Through application of the solution of the present disclosure, a scalable and learnable large model augmentation framework having multiple component capabilities is proposed. Based on the task processing model, the integration of multiple artificial intelligence models through processing components can, on the one hand, enhance a text generation capability of the task processing model in natural language tasks, and on the other hand, compensate for lack of multi-modal generation capabilities of the task processing model, so that a processing capability of the task processing model is more comprehensive.

Referring to FIG. 9, FIG. 9 is an example diagram of a task in a task processing method according to some embodiments of the present disclosure. As shown in FIG. 9, a task processing platform provided in the present disclosure supports different types of tasks, including but not limited to, a video generation task, a picture generation task, a text-to-image conversation task, a speech synthesis task, a knowledge question answering task, an address structuring task, an English-Chinese translation task, a named entity recognition task, a Chinese-English translation task, and a general information extraction task. Next, a question answering process of each task is described in detail one by one.

Video generation task: Question 1: Generate a video of two puppies playing. Answer 1: Currently, the task processing platform has a video generation component supporting English, thus I will first convert your question into English through a Chinese-English translation model, and then call an English video generation model, to obtain a translation of the question

that is Two puppies playing, and a video is being generated through the video generation model (the video displayed).

Picture generation task: Question 1: Generate a high-definition picture of a puppy lying on a sofa. Answer 1: Call a picture generation component for generation (the picture displayed).

Text-to-image conversation task: Question 1: What color is the clothes of a man in a picture **A**. Answer 1: Yellow.

Speech synthesis task: Question 1: Write a 16-word environmental protection slogan, and read it out in a female voice. Answer 1: Green mountains and clear waters are our home; let's protect them together! (read the slogan in a female voice).

Knowledge question answering task: Question 1: What is the characteristic of a model **A**. Answer 1: Support efficient input of texts such as external knowledge and persona information for knowledge enhancement.

Address structuring task: Question 1: Parse the following address, and extract specific elements: Room 401, No. 56, Lane E, Residential Quarter D, Sub-district C, District B, City A, 11111111111x. Answer 1: Address breakdown in JSON format: {"city": "City A", "district": "District B", "houseno": "No. 56", "poi": "Residential Quarter D", "road": "E", and "town": "Sub-district C"}. The model I use is a model XX. This is obtained through fine-tuning based on a multi-task multi-modal address pre-trained foundation model.

Named entity recognition task: Question 1: Identify entities in the following sentence: The 2023 Artificial Intelligence Conference was held in City M. Answer 1: Entity extraction result in JSON format: {"GPE": ["City M"]}. The model I use is a model XXX.

English-Chinese translation task: Question 1: Translate the following sentence into Chinese: I love learning. Answer: 我爱学习.

Chinese-English translation task: Question 1: Translate the following sentence into English: 今天天气真好. Answer: The weather is so nice today.

General information extraction task: Question 1: Extract information corresponding to the following text based on a given schema, schema: {"Person": null, "Geographic Location": null, "organization": null, "Time": null}, in 2019, University A held the sixth Forum in Location B.

More than 200 young scientists from around the world participated in this forum, including Li Si, Wang Wu, and the like. Answer 1: Information extraction result: {“Person”: [“Li Si”, “Wang Wu”], “Geographic Location”: [“Location B”], “Time”: [“2019”], and “Organization”: [“University **A**”]}. The model I use is a model XX. This is a general information extraction model.

Referring to FIG. 10, FIG. 10 is a schematic interface diagram of an automatic question answering interface according to some embodiments of the present disclosure. The automatic question answering interface includes a question processing request input interface and a reply result display interface. The question processing request input interface includes a question processing request input box, an “OK” control, and a “Cancel” control. The reply result display interface includes a reply result display box.

A user inputs a question processing request through the question processing request input box displayed by a client, where the question processing request carries a question to be answered, and clicks on the “OK” control, and the task processing platform receives the question processing request sent by the client, and determines a question to be answered and component description information into a task processing model to determine at least one target processing component corresponding to the question processing request. The task processing model is a deep learning model. The at least one target processing component is called to process the question to be answered to obtain a reply result corresponding to the question processing request, and the reply result is sent to the client. The client displays the reply result in the reply result display box. In a practical application, the user operates a control in a manner including any one of click, double-click, touch, mouse hover, swipe, long press, voice control, shake, or the like, which is specifically selected based on an actual situation. The present disclosure does not impose any limitation on this.

Corresponding to the embodiments of the task processing method, some embodiments of the present disclosure further provide a task processing apparatus. FIG. 11 is a structural schematic diagram of a task processing apparatus according to some embodiments of the present disclosure. As shown in FIG. 11, the task processing apparatus is applied to a service unit in a task processing platform. The task processing platform includes the service unit and multiple processing components. The task processing apparatus includes:

a first receiving module 1102, configured to receive a task processing request, where the task processing request carries data to be processed;

a first input module 1104, configured to input the data to be processed and component description information into a task processing model to determine at least one target processing component corresponding to the task processing request, where the task processing model is a deep learning model; and

a first calling module 1106, configured to call the at least one target processing component to process the data to be processed, to obtain a task processing result corresponding to the task processing request.

Optionally, the apparatus further includes: a matching module, configured to obtain reference text data; and match the data to be processed with the reference text data to obtain a matching result. The first input module 1104 is further configured to, in response to the matching result indicating a matching failure, input the data to be processed and the component description information into the task processing model to determine the at least one target processing component corresponding to the task processing request.

Optionally, the matching module is further configured to, in response to the matching result indicating a matching success, input the data to be processed and the reference text data into the task processing model to determine the task processing result corresponding to the task processing request.

Optionally, the apparatus further includes: a training module, configured to obtain a sample set, where the sample set includes the multiple samples, each sample carries a sample label, and each sample is constructed based on at least one of sample text data, sample question-answer pair data, and sample component call data; extract a first sample from the sample set, where the first sample is any one of the multiple samples; input the first sample into the task processing model to obtain a first prediction result of the first sample; calculate a first loss value based on the first prediction result and a first sample label carried in the first sample; and adjust model parameters of the task processing model based on the first loss value, and return to execute an operation of extracting the first sample from the sample set, until a first preset stopping condition is reached to obtain the task processing model resulting from the training method.

Optionally, the training module is further configured to parse the first prediction result to obtain multiple first prediction sub-results; determine a weight corresponding to each of the first prediction sub-results based on a type of each first prediction sub-result; and calculate the first loss value based on the multiple first prediction sub-results, the weight corresponding to each first prediction sub-result, and the first sample label carried in the first sample.

Optionally, the apparatus further includes: a conversion module, configured to obtain component call parameters of at least one target processing component; and perform format conversion on the data to be processed based on the component call parameters to obtain updated data to be processed. The first calling module 1106 is further configured to call the at least one target processing component to process the updated data to be processed to obtain the task processing result corresponding to the task processing request.

Optionally, the apparatus further includes: an optimization module, configured to obtain call information of the at least one target processing component, and optimize the task processing model based on the call information.

Optionally, the task processing request further carries task requirement information. The apparatus further includes: a third input module, configured to input the task requirement information and the task processing result into the task processing model to obtain an updated task processing result, where the updated task processing result is consistent with the task requirement information.

Optionally, the apparatus further includes: a sending module, configured to send the task processing result to a client, so that the client displays the task processing result to a user.

Optionally, the apparatus further includes: a second receiving module, configured to receive a post-processing request sent by the user based on the task processing result, and process the post-processing request to obtain a post-processing result corresponding to the post-processing request.

Through application of the solution of the present disclosure, the multiple processing components are integrated on the task processing platform, so that the at least one target processing component is directly called to process the data to be processed, thereby improving task processing efficiency. In addition, the multiple processing components can, on the one

hand, enhance a text generation capability of the task processing model in natural language tasks, and on the other hand, compensate for lack of multi-modal generation capabilities of the task processing model, so that a processing capability of the task processing model is more comprehensive.

The above is a schematic solution of the task processing apparatus of some embodiments. It is to be noted that the technical solution of the task processing apparatus and the technical solution of the task processing method belong to the same concept. For details not described in detail in the technical solution of the task processing apparatus, reference is made to the description of the technical solution of the task processing method.

Corresponding to the embodiments of the automatic question answering method, some embodiments of the present disclosure further provide an automatic question answering apparatus. FIG. 12 is a structural schematic diagram of an automatic question answering apparatus according to some embodiments of the present disclosure. As shown in FIG. 12, the automatic question answering apparatus is applied to a service unit in a task processing platform. The task processing platform includes the service unit and multiple processing components. The automatic question answering apparatus includes:

a first obtaining module 1202, configured to obtain a question to be answered;

a second input module 1204, configured to input the question to be answered and component description information into a task processing model to determine at least one target processing component corresponding to the question to be answered, where the task processing model is a deep learning model for natural language;

a second calling module 1206, configured to call the at least one target processing component to process the question to be answered, to obtain a reply result corresponding to the question to be answered, and send the reply result to a user; and

a determining module 1208, configured to receive an updated question to be answered sent by the user based on the reply result, and perform the step of inputting the question to be answered and the component description information into the task processing model to determine the at least one target processing component corresponding to the question to be answered, until a second preset stop condition is satisfied, to obtain the reply result.

Through application of the solution of the present disclosure, the multiple processing components are integrated on the task processing platform, so that the at least one target processing component is directly called to process the question to be answered, thereby improving automatic question answering efficiency. In addition, the multiple processing components can, on the one hand, enhance a text generation capability of the task processing model in natural language tasks, and on the other hand, compensate for lack of multi-modal generation capabilities of the task processing model, so that the automatic question answering is more comprehensive.

The foregoing is a schematic solution of the automatic question answering apparatus of some embodiments. It is to be noted that the technical solution of the automatic question answering apparatus and the technical solution of the automatic question answering method belong to the same concept. For details not described in detail in the technical solution of the automatic question answering apparatus, reference is made to the description of the technical solution of the automatic question answering method.

FIG. 13 is a structural block diagram of a computing device according to some embodiments of the present disclosure. Components of the computing device 1300 include but are not limited to a memory 1310 and a processor 1320. The processor 1320 is connected with the memory 1310 through a bus 1330, and a database 1350 is configured to store data.

The computing device 1300 further includes an access device 1340. The access device 1340 enables the computing device 1300 to perform communication through one or more networks 1360. Examples of the networks include a public switched telephone network (PSTN), a local area network (LAN), a wide area network (WAN), a personal area network (PAN), or a combination of communication networks such as the Internet. The access device 1340 includes at least one of any type of wired or wireless network interfaces (for example, a network interface card (NIC)), such as an IEEE802.11 wireless local area network (WLAN) wireless interface, a worldwide interoperability for microwave access (Wi-MAX) interface, an Ethernet interface, a universal serial bus (USB) interface, a cellular network interface, a Bluetooth interface, and a near field communication (NFC) interface.

In some embodiments of the present disclosure, the components of the computing device 1300 and other components not shown in FIG. 13 are connected with each other, for example,

through a bus. It should be understood that the structural block diagram of the computing device shown in FIG. 13 is used for illustration, and is not a limitation on the scope of the present disclosure. A person having ordinary skill in the art adds other components or replaces the components with other components as required.

The computing device 1300 is any type of stationary or mobile computing device, including a mobile computer or a mobile computing device (such as a tablet computer, a personal digital assistant, a laptop computer, a notebook computer, or a netbook), a mobile phone (such as a smartphone), a wearable computing device (such as a smartwatch or smart glasses), or another type of mobile device, or a stationary computing device such as a desktop computer or a personal computer (PC). The computing device 1300 is a stationary or mobile server.

The processor 1320 is configured to execute the following computer-executable instructions. When the computer-executable instructions are executed by the processor, the steps of the task processing method are implemented.

The above is an exemplary solution of the computing device in some embodiments. It is to be noted that the technical solution of the computing device and the technical solution of the task processing method belong to the same concept. For details not described in detail in the technical solution of the computing device, reference is made to the description of the technical solution of the task processing method.

Some embodiments of the present disclosure further provide a computer-readable storage medium, storing computer-executable instructions. The computer-executable instructions, when executed by a processor, implement the steps of the task processing method.

The above is a schematic solution of the computer-readable storage medium of this embodiment. It is to be noted that the technical solution of the storage medium and the technical solution of the task processing method belong to the same concept. For details not described in detail in the technical solution of the storage medium, reference is made to the description of the technical solution of the task processing method.

Some embodiments of the present disclosure further provide a computer program. The computer program, when executed in a computer, causes the computer to perform the steps of the task processing method.

The above is an exemplary solution of the computer program in this embodiment. It is to be noted that the technical solution of the computer program and the technical solution of the task processing method belong to the same concept. For details not described in detail in the technical solution of the computer program, reference is made to the description of the technical solution of the task processing method.

Specific embodiments of the present disclosure are described above. Other embodiments fall within the scope of the appended claims. In some cases, the actions or steps described in the claims are performed in a sequence different from that in the embodiments, and desired results are still achieved. In addition, the processes depicted in the accompanying drawings are not necessarily performed in a specific order or sequential order to achieve the desired results. In some implementations, multitasking and parallel processing are also possible or are advantageous.

The computer instruction includes computer program code. The computer program code is in the form of source code, object code, and an executable file, some intermediate forms, or the like. The computer-readable medium includes: any entity or apparatus capable of carrying the computer program code, a recording medium, a USB flash disk, a mobile hard disk, a magnetic disk, an optical disk, a computer memory, a read-only memory (ROM), a random access memory (RAM), an electric carrier signal, a telecommunication signal, a software distribution medium, and the like.

It should be noted that for brief description, the method embodiments are described as a series of action combinations. However, it is to be appreciated by a person having ordinary skill in the art that the present disclosure is not limited to the described sequence of the actions, because some steps are performed in another sequence or simultaneously according to the present disclosure. In addition, it is to be appreciated by a person having ordinary skill in the art that the embodiments described in the specification all belong to exemplary embodiments, and the involved actions and modules are not necessary for the present disclosure.

In the embodiments, the descriptions of the embodiments have respective emphasis. For a part that is not described in detail in some embodiments, reference is made to the related descriptions of other embodiments.

The exemplary embodiments of the present disclosure disclosed above are intended to assist in describing the present disclosure. The optional embodiments do not describe all details in detail, and do not limit the present invention to being the specific implementations. Apparently, many modifications and changes are made according to the contents of the present disclosure. The embodiments are selected and described in detail in the present disclosure, to better explain the principles and practical applications of the present disclosure, so that a person having ordinary skill in the art can better understand and use the present disclosure. The present disclosure is limited by the claims and the full scope and equivalents thereof.

Industrial applicability

The task processing method provided in the embodiments of the present disclosure is applied to a service unit in a task processing platform. The task processing platform includes the service unit and multiple processing components. A task processing request is received, where the task processing request carries data to be processed. The data to be processed and component description information are inputted into a task processing model, and at least one target processing component corresponding to the task processing request is determined, where the task processing model is a deep learning model. The at least one target processing component is called to process the data to be processed to obtain a task processing result corresponding to the task processing request. The multiple processing components are integrated on the task processing platform, so that the at least one target processing component is directly called to process the data to be processed, thereby improving task processing efficiency. In addition, the multiple processing components can, on the one hand, enhance a text generation capability of the task processing model in natural language tasks, and on the other hand, compensate for lack of multi-modal generation capabilities of the task processing model, so that a processing capability of the task processing model is more comprehensive.

Claims

1. A task processing method, applied to a service unit in a task processing platform, wherein the task processing platform comprises the service unit and a plurality of processing components, and the task processing method comprises:

   receiving a task processing request, wherein the task processing request carries data to be processed;

   inputting the data to be processed and component description information into a task processing model to determine at least one target processing component corresponding to the task processing request, wherein the task processing model is a deep learning model; and

   calling the at least one target processing component to process the data to be processed, to obtain a task processing result corresponding to the task processing request.

2. The task processing method according to claim 1, wherein before inputting the data to be processed and the component description information into the task processing model to determine the at least one target processing component corresponding to the task processing request, the task processing method further comprises:

   obtaining reference text data; and

   matching the data to be processed with the reference text data to obtain a matching result; and

   inputting the data to be processed and the component description information into the task processing model to determine the at least one target processing component corresponding to the task processing request comprises:

   in response to the matching result indicating a matching failure, inputting the data to be processed and the component description information into the task processing model to determine the at least one target processing component corresponding to the task processing request.

3. The task processing method according to claim 2, wherein after matching the data to be processed with the reference text data to obtain the matching result, the task processing method further comprises:

in response to the matching result indicating a matching success, inputting the data to be processed and the reference text data into the task processing model to determine the task processing result corresponding to the task processing request.

4. The task processing method according to claim 1, wherein a training manner of the task processing model comprises:

obtaining a sample set, wherein the sample set comprises the plurality of samples, each sample carries a sample label, and each sample is constructed based on at least one of sample text data, sample question-answer pair data, and sample component call data;

extracting a first sample from the sample set, wherein the first sample is any one of the plurality of samples;

inputting the first sample into the task processing model to obtain a first prediction result of the first sample;

calculating a first loss value based on the first prediction result and a first sample label carried in the first sample; and

adjusting model parameters of the task processing model based on the first loss value, and returning to execute an operation of extracting the first sample from the sample set, until a first preset stopping condition is reached to obtain the task processing model resulting from the training method.

5. The task processing method according to claim 4, wherein calculating the first loss value based on the first prediction result and the first sample label carried in the first sample comprises:

parsing the first prediction result to obtain a plurality of first prediction sub-results;

determining a weight corresponding to each of the first prediction sub-results based on a type of each of the first prediction sub-results; and

calculating the first loss value based on the plurality of first prediction sub-results, the weight corresponding to each of the first prediction sub-results, and the first sample label carried in the first sample.

6. The task processing method according to claim 1, wherein before calling the at least one target processing component to process the data to be processed, to obtain the task processing result corresponding to the task processing request, the task processing method further comprises:

obtaining component call parameters of the at least one target processing component; and performing format conversion on the data to be processed based on the component call parameters to obtain updated data to be processed; and

calling the at least one target processing component to process the data to be processed, to obtain the task processing result corresponding to the task processing request comprises:

calling the at least one target processing component to process the updated data to be processed, to obtain the task processing result corresponding to the task processing request.

7. The task processing method according to claim 1, wherein after calling the at least one target processing component to process the data to be processed, to obtain the task processing result corresponding to the task processing request, the task processing method further comprises:

obtaining call information of the at least one target processing component, and optimizing the task processing model based on the call information.

8. The task processing method according to claim 1, wherein the task processing request further carries task requirement information; and

after calling the at least one target processing component to process the data to be processed, to obtain the task processing result corresponding to the task processing request, the task processing method further comprises:

inputting the task requirement information and the task processing result into the task processing model to obtain an updated task processing result, wherein the updated task processing result is consistent with the task requirement information.

9. The task processing method according to any one of claims 1 to 8, wherein after calling the at least one target processing component to process the data to be processed, to obtain the task processing result corresponding to the task processing request, the task processing method further comprises:

sending the task processing result to a client, so that the client displays the task processing result to a user.

10. The task processing method according to claim 9, wherein after sending the task processing result to the client, the task processing method further comprises:

receiving a post-processing request sent by the user based on the task processing result, and processing the post-processing request to obtain a post-processing result corresponding to the post-processing request.

11. The task processing method according to claim 1, wherein inputting the data to be processed and the component description information into the task processing model to determine the at least one target processing component corresponding to the task processing request comprises:

screening a plurality of processing components in the task processing platform to determine at least one candidate processing component;

inputting the data to be processed and the component description information of the at least one candidate processing component into the task processing model to determine the at least one target processing component corresponding to the task processing request;

wherein screening the plurality of processing components in the task processing platform to determine the at least one candidate processing component comprises:

comparing a task type of the task processing request with a component type of each processing component, to screen out the at least one candidate processing component matching the task type from the plurality of processing components; or,

comparing task description information of the task processing request with component function information of each processing component, to screen out the at least one candidate processing component matching the task description information from the plurality of processing components.

12. A task processing system, comprising a communication interface and a calling unit;

the communication interface, configured to receive a processing request sent by a client, and forward the processing request to the calling unit, wherein the processing request carries data to be queried; and

the calling unit, configured to, in response to the processing request, call a query model to process the data to be queried and processing model description information to determine at least one target processing model corresponding to the data to be queried, wherein the query model is a deep learning model; and send model information of the at least one target processing model to the client through the communication interface.

13. The system according to claim 12, wherein the communication interface is further configured to receive model adjustment parameters sent by the client, and forward the model adjustment parameters to the calling unit; and

    the calling unit is further configured to adjust the at least one target processing model based on the model adjustment parameters, and send model information of at least one adjusted target processing model to the client through the communication interface.

14. A computing device, comprising:

    a memory and a processor;

    wherein the memory is configured to store computer-executable instructions, the processor is configured to execute the computer-executable instructions, and the computer-executable instructions, when executed by the processor, implement steps of the task processing method as claimed in any one of claims 1 to 11.

15. A computer-readable storage medium, storing computer-executable instructions, wherein the computer-executable instructions are executed by a processor to implement steps of the task processing method as claimed in any one of claims 1 to 11.