LARGE MODEL TRAINING METHOD AND APPARATUS, AND ENTITY INFORMATION EXTRACTION METHOD AND APPARATUS

TECHNICAL FIELD

[0001] The present disclosure relates to the field of image processing technologies, and in particular, to a large model training method and apparatus, and an entity information extraction method and apparatus.

BACKGROUND

[0002] In routine work of technical support and operation and maintenance, processing various words and images used to express an abnormality in a problem work order, form, or order and extracting entity information from the problem work order, form, or order to query related information in an internal link or system log to locate a problem are essential steps.

[0003] An existing problem work order, form, or order is mainly processed based on manual entry and troubleshooting of a frontline person. However, some information in various words and images is lengthy and error-prone, for example, a digital ID, and consequently, efficiency improving of a processing process is hindered.

[0004] With development and application of a large model, the large model starts to be introduced in abnormality troubleshooting of the problem work order, form, or order, to form an intelligent customer service system. However, most of current large models are based on a language model, are only applicable to processing word information, and cannot directly process image information. This limits widespread application of a large model technology in the field of abnormality troubleshooting of the problem work order, form, or order.

SUMMARY

[0005] The present disclosure provides a large model training method, an entity information extraction method, a computer system, and a storage device.

[0006] According to a first aspect, a large model training method is provided, including: constructing a training set, where the training set includes at least a first-type sample, input data of the first-type sample includes an OCR output obtained by performing text recognition on an original image by using an OCR model, the OCR output includes a recognized text element and location information of each element, and label data of the first-type sample includes entity information in the original image; and performing fine-tuning on a pretrained large language model based on the training set.

[0007] According to a second aspect, an entity information extraction method is provided, including: obtaining a fine-tuned large language model according to any one of claims 1 to 9; performing text recognition on a target image by using an OCR model, to obtain a target OCR output, where the target OCR output includes a recognized text element and location information of each text element; constructing a prompt based on the target OCR output, where the prompt indicates an entity recognition task; and inputting the prompt into the large language model, to obtain entity information output by the large language model.

[0008] According to a third aspect, a computer system is provided, including a storage, a processor, and a computer program that is stored in the storage and that is capable of running on the processor. The processor performs the following target operation for at least one time when executing the program, and implements the following steps by performing the target operation: constructing a training set, where the training set includes at least a first-type sample, input data of the first-type sample includes an OCR output obtained by performing text recognition on an original image by using an OCR model, the OCR output includes a recognized text element and location information of each element, and label data of the first-type sample includes entity information in the original image; and performing fine-tuning on a pretrained large language model based on the training set.

[0009] According to a fourth aspect, a computer system is provided, including a storage, a processor, and a computer program that is stored in the storage and that is capable of running on the processor. The processor performs the following target operation for at least one time when executing the program, and implements the following steps by performing the target operation: obtaining a fine-tuned large language model according to any one of claims 1 to 9; performing text recognition on a target image by using an OCR model, to obtain a target OCR output, where the target OCR output includes a recognized text element and location information of each text

element; constructing a prompt based on the target OCR output, where the prompt indicates an entity recognition task; and inputting the prompt into the large language model, to obtain entity information output by the large language model.

[0010] According to a fifth aspect, a computer-readable storage medium is provided. The storage medium stores a computer program, and when the computer program is executed by a processor, the method according to any implementation of the first aspect or the second aspect is implemented.

[0011] The technical solutions provided in the embodiments of this specification can include the following beneficial effects.

[0012] An OCR output of an original image is obtained by using an OCR model, a training set for an entity information extraction task is constructed based on the OCR output, and a pretrained large language model is fine-tuned based on the constructed training set, so that the fine-tuned large language model has a capability of extracting entity information in the original image based on the OCR output, to improve abnormality troubleshooting efficiency of a problem work order, form, or order by using the large model.

[0013] It should be understood that the above-mentioned general description and the following detailed description are merely examples and explanations, and should not limit this application.

BRIEF DESCRIPTION OF DRAWINGS

[0014] To describe the technical solutions in embodiments of this specification more clearly, the following briefly describes the accompanying drawings needed for describing the embodiments. Clearly, the accompanying drawings in the following descriptions show merely some embodiments recorded in this specification, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0015] FIG. 1 is a schematic diagram of an original image processing solution according to an example embodiment of this specification;

[0016] FIG. 2 is a flowchart of a large model training method according to an example embodiment of this specification;

[0017] FIG. 3 is a schematic diagram of fine-tuning of a large language model according to an example embodiment of this specification;

[0018] FIG. 4 is a flowchart of an entity information extraction method according to an example embodiment of this specification; and

[0019] FIG. 5 is a schematic diagram of a computer system according to an example embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

[0020] To make a person skilled in the art better understand the technical solutions of this specification, the following clearly and completely describes the technical solutions in the embodiments of this specification with reference to the accompanying drawings in the embodiments of this specification. Clearly, the described embodiments are merely some rather than all of the embodiments of this specification. All other embodiments obtained by a person of ordinary skill in the art based on the embodiment of this specification without creative efforts shall fall within the protection scope of this specification.

[0021] When the following description relates to the accompanying drawings, unless specified otherwise, same numbers in different accompanying drawings represent same or similar elements. Implementations described in the following example embodiments do not represent all implementations consistent with the present disclosure. On the contrary, the embodiments are merely examples of apparatuses and methods that are described in the appended claims in details and consistent with some aspects of the present disclosure.

[0022] The terms used in the present disclosure are merely for illustrating specific embodiments, and are not intended to limit the present disclosure. The terms "a", "said", and "the" of singular forms used in the present disclosure are also intended to include plural forms, unless otherwise specified in the context clearly. It should be further understood that the term "and/or" used in this specification indicates and includes any or all possible combinations of one or more associated listed items.

[0023] It should be understood that although terms "first", "second", "third", etc. may be used in the present disclosure to describe various types of information, the information is not limited to the terms. These terms are used merely to differentiate information of the same type. For example, without departing from the scope of the present disclosure, first information can also be referred to as second information, and similarly, the second information can be referred to as the first

information. Depending on the context, for example, the word "if" used here can be explained as "while", "when", or "in response to determining".

[0024] As described above, in addition to word information, various original images expressing an abnormality are included in a problem work order, form, or order. The original images include a large amount of entity information, and the information is also critical to a subsequent abnormality troubleshooting process.

[0025] In some solutions, key information/entity information can be extracted from image information of the work order and form in a conventional rule method. Specifically, various elements in the original image and coordinates of the elements are recognized by using an optical character recognition (OCR) model, and then row and column information of coordinates in a case of a specific pixel difference based on result features recognized by using the OCR model and with reference to various manually predefined rules, so that the various elements are combined based on the row and column information, to obtain the entity information in the original image. However, when the OCR model is updated or replaced, a recognized result and precision usually change, various rules need to be updated, and a large amount of manpower is needed. In addition, such processing requires relatively strong expert experience. Recognition of new images and entity information is inaccurate and inflexible.

[0026] With rapid development of a large model, more models are involved in processing of a problem work order, form, or order. However, a common large language model (LLM) in a current large model can only process and understand word information, and cannot directly process image information. This limits widespread application of the large language model in the field of abnormality troubleshooting of the problem work order, form, or order.

[0027] In some other solutions, both text information and image information included in the problem work order, form, or order can be processed by using a multimodal large model (MLM) that supports an image input. However, a huge investment in training, from the beginning, a large multimodal large model that supports an image input in a target field is mainly reflected as a large quantity of computing and storage resources that need to be consumed and a large quantity of professional persons that need to be involved. In addition, the multimodal large model has a long period from development to final deployment, and is difficult to be landed, to further increase time costs. In addition, the multimodal large model is usually developed for the target field, and is obtained through training based on a large quantity of labeling samples of an existing type of entity

information. When a new type of entity information needs to be extracted from a processed image, performance is usually poor. Therefore, relatively high availability cannot be achieved, and consequently, generalization is poor.

[0028] In addition, currently, no common multimodal large model base with relatively high availability is disclosed, and therefore, the common multimodal large model base cannot be directly integrated based on a labeling sample for the target field to perform fine-tuning development.

[0029] These factors limit widespread application of the multimodal large model in the field of abnormality troubleshooting of the problem work order, form, or order.

[0030] In view of this, a solution is proposed in this specification. An OCR output of an original image is obtained by using an OCR model, a training set for an entity information extraction task is constructed based on the OCR output, and a pretrained large language model is fine-tuned based on the constructed training set, so that the fine-tuned large language model has a capability of extracting entity information in the original image based on the OCR output. In an entity information extraction phase, the OCR model and the fine-tuned large language model are combined in series, to improve entity information extraction efficiency of the original image, thereby improving abnormality troubleshooting efficiency of the problem work order, form, or order.

[0031] The following describes in detail an implementation of the solution.

[0032] For example, FIG. 1 is a schematic diagram of an original image processing solution according to an example embodiment. As shown in FIG. 1, the processing solution includes the following content.

[0033] Step S101: Obtain a plurality of original images.

[0034] Abnormality troubleshooting of a problem work order is used as an example. A worker can sort various types of problem work orders provided by an internal and external partner or user, to obtain a plurality of original images that need to be processed in the problem work order. The internal and external partner or user can be a user or development participant of a related product or technology in a target field, and the worker can be a maintenance person who counts the product or technology.

[0035] The original image can be a problem/interface screenshot when the product or technology becomes abnormal. A specific format of the original image is not limited, for example,

jpg or png. The original image includes entity information used for abnormality troubleshooting of the problem work order. The entity information is a specific object included in the original image, for example, a user ID, account password information, a transaction document, a company name, and an organization code. To subsequently construct a training set, the entity information in the original image can be labeled, to obtain a label of the entity information. The label of the entity information can include entity content and a label of an entity type.

[0036] Step S102: Perform text recognition on the original image by using an OCR model.

[0037] The OCR model is used to convert a text image area in the original image into an editable and searchable text element. The model can be an algorithm model self-developed for the target field, or can be a mature publicly released tool library, for example, Tesseract OCR or EasyOCR, provided that a selected model can reach precision required by a service. The text element is a character string in the original image, and can include a text in any language, for example, a Chinese word, an English word, and a meaningful symbol, for example, an Arabic number or a unit of measurement symbol.

[0038] The plurality of obtained original images are sequentially input into the OCR model, and text recognition is separately performed on the input original images by using the OCR model, to obtain OCR outputs corresponding to the original images. The OCR output specifically includes a recognized text element and location information of each element. The location information is information about an area occupied by each element in the image, and is usually represented by using coordinates, for example, is represented as vertex coordinates of a rectangle corresponding to the occupied area.

[0039] Step S103: Construct the training set.

[0040] To extract the entity information in the image, a training set used to train a large language model is constructed. The training set includes at least a training sample used for an entity extraction task. Input data includes the obtained OCR output, and label data includes the entity information labeled for the original image.

[0041] It can be understood that, because providers and generation environments of the problem work orders are different, an original image expressing an abnormality in the problem work order is non-standard and irregular, so that directly extracting the entity information based on the OCR output becomes complex. For example, because a length-to-width ratio, resolution, or a display angle of a text image area in the original image do not meet a specification, an

information unit with a left-right/up-down structure is distributed in left and right columns or up and down rows and therefore is recognized as different text elements. Alternatively, because there is a line break in a text image in the original image, entity information is distributed in up and down rows and therefore is recognized as different text elements. Alternatively, a text image includes a modification, painting, or a special symbol, the recognized text element includes noise and a scrambling code. When entity information is directly extracted based on an OCR output of such a non-standard and irregular image, a task becomes relatively difficult.

[0042] Therefore, it is considered to perform a plurality of types of preprocessing on data included in the OCR output, and correct an error caused by a non-standard and irregular case, to facilitate subsequent entity information extraction. A training set can also be correspondingly constructed for this part of preprocessing task, so that such preprocessing can be implemented after the large language model is fine-tuned.

[0043] Specifically, a preprocessing task can be set for data cleaning and verification. A training sample correspondingly constructed for the task can be as follows: Input data is an OCR output, label data is a text element after data cleaning is performed on the OCR output, and data cleaning includes: clearing noise and a scrambling code. A preprocessing task can also be set, to recognize row and column information of the element in the OCR output. A training sample correspondingly constructed for the task can be as follows: Input data is an OCR output, label data is elements in the same row, and/or elements in the same column.

[0044] The label data can be generated through manual labeling, or can be generated by using a semi-automatic tool in an assisted manner.

[0045] Step S104: Construct a prompt.

[0046] The prompt is used to determine a specific task or target completed by the large language model in a fine-tuning or application process, to improve recognition efficiency of the entity information in the original image.

[0047] Optionally, a valid prompt can be designed based on an expected entity information output format, to guide the large language model to output the entity information in the original image based on a requirement of the prompt, so as to further improve recognition efficiency of the entity information.

[0048] The prompt can be constructed based on the OCR output obtained by processing the original image based on the OCR model.

[0049] Step S105: Perform fine-tuning on a pretrained large language model based on the training set.

[0050] The pretrained large language model can be a model self-developed for the target field, or can be a mature publicly released general model. The large language model is used to perform entity extraction based on the OCR output, and an output result of the model includes entity information of an expected extraction type and in an expected output format in the original image.

[0051] When the large language model is fine-tuned, the pretrained large language model is used as a base, the constructed prompt is input into the large language model to obtain a predicted output, and a parameter of the large language model is adjusted based on the predicted output and the label data of the sample, so that the large language model adapts to the above-mentioned plurality of preprocessing tasks and the above-mentioned plurality of entity information extraction tasks.

[0052] Step S106: Extract entity information by using the fine-tuned large language model.

[0053] When the fine-tuned large language model is applied, the original image is input into the OCR model to obtain the OCR output, the prompt is constructed by using the OCR output, and the prompt is provided to the fine-tuned large language model for entity extraction.

[0054] Step S107: Output the entity information.

[0055] The entity information in the original image is obtained and output based on a requirement of the prompt.

[0056] It can be understood that an execution sequence of steps S103 to S106 can be adjusted based on an actual situation, and is not limited here.

[0057] Therefore, when fine-tuning training is performed on the large language model, preliminary character recognition is performed on the original image by using an existing mature OCR model, to obtain the OCR output, the training set is constructed based on the OCR output and label data of an entity extraction related task, and fine-tuning training is performed on the large language model based on the training set, so that the large language model adapts to the plurality of pre-processing tasks and the plurality of entity information extraction tasks. Even if the OCR model is updated or replaced, a large language model that can obtain the entity information in the original image based on the OCR output can be developed in a relatively small amount of time and at low technical costs. In this way, development costs of a large model that can be widely used in the field of abnormality troubleshooting of a problem work order, form, or order are reduced,

and development duration is reduced.

[0058] In addition, the OCR model and the fine-tuned large language model are combined in series, and the entity information in the original image can be obtained based on the OCR output by using the fine-tuned large language model, to improve extraction efficiency of the entity information in the original image.

[0059] In addition, the valid prompt is designed based on an expected processing task or an expected output format, and the large language model is guided to extract the entity information in the original image based on a requirement of the prompt and output the entity information, to improve abnormality troubleshooting efficiency of the problem work order.

[0060] Based on the above-mentioned content, FIG. 2 is a flowchart of a large model training method according to an example embodiment. The method can be performed by using a computing platform, and the computing platform can be embodied as any computing unit, server, device, device cluster, etc. that has a computing and processing capability. The large model training method includes the following steps.

[0061] Step S201: Construct a training set, where the training set includes at least a first-type sample, input data of the first-type sample includes an OCR output obtained by performing text recognition on an original image by using an OCR model, the OCR output includes a recognized text element and location information of each element, and label data of the first-type sample includes entity information in the original image.

[0062] In an embodiment, the original image is an image in a to-be-processed problem work order, and includes a large amount of entity information for root cause analysis. The original image is usually used to express an abnormality, and can be an abnormality page screenshot in various forms and formats. The training set is used to provide basic data for learning for a large language model.

[0063] The OCR output is an output result obtained by performing text recognition on the original image by using various OCR models. The output result can be a plaintext or a structured format, for example, XML or JSON, for further processing and analysis. The OCR output includes a recognized text element and location information of each element.

[0064] For example, an OCR output in a JSON format includes the following content:

[OCRData(word=Transaction ID: 2023-XXXX-1232-1234◢◣-2231, rect={top=551.0, left=65.0, width=203.0, angle=0.0, height=31.0}), OCRData(word=[sPay,

rect={top=338.0, left=485.0, width=56.0, height=118.0}), OCRData(word=See more, rect={top=1396.0, left=551.0, width=138.0, angle=0.0, height=32.0}), OCRData(word=Transfer, rect={top=426.0, left=72.0, width=100.0, angle=0.0, height=26.0}), OCRData(word=PA-ALIEXPRESS LOC: 8Pending Nov 30, DemandPending, rect={top=1010.0, left=70.0, width=325.0, height=67.0})]

[0065] The field word represents the recognized text element. For example, "word=Transfer" represents that the recognized text element is "Transfer". The location information of the text element is usually represented in a form of coordinates in the image. According to different OCR models, the location information of each element can be four coordinate values (usually, horizontal and vertical coordinate values of an upper left corner and a lower right corner of a rectangle), or can be two coordinate values and length and width information of the rectangle represented by the field rect. For example, top=551.0, left=65.0, width=203.0, angle=0.0, and height=31.0. Here, top and left jointly form coordinates of the upper left corner of the rectangle, width and height represent the length and width information of the rectangle, and angle is optional and represents a rectangular tilt angle.

[0066] In the following, the training set is constructed based on the OCR output described above.

[0067] In an example, the first-type sample in the training set is constructed, the input data of the first-type sample is the OCR output, and the label data includes the entity information in the original image.

[0068] Specifically, a small quantity (for example, one to two thousand samples) of original images can be collected. First, the OCR model is used to separately perform text recognition on the original images, to obtain OCR outputs corresponding to the original images. These OCR outputs are used as input data for the first-type sample. Then, entity information in the original image or the OCR output corresponding to the original image is labeled, to obtain a label of the entity information. The label of the entity information can include entity content and a label of an optional entity type.

[0069] For example, if an original image includes a text: "The XYZ host is automatically restarted at 8:00 p.m. on the New Year's Day in 2021." The entity information label obtained through labeling may include:

Entity content: XYZ host, 8:00 p.m. on the New Year's Day in 2023, automatically restarted

Entity type: Device NAME, DATE, EVENT

[0070] In an actual labeling process, the information is usually stored in a form of structured data, for example, JSON or XML. Stored label data includes the entity information in the original image.

[0071] The input data and the label data are used to subsequently perform fine-tuning training on the pretrained large language model, so that the fine-tuned large language model completes a task of extracting the entity information in the original image based on the OCR output. The first-type sample includes input data and label data in this instance.

[0072] In another example, a second-type sample in the training set is further constructed. Input data of the second-type sample is the OCR output, and label data is text elements in the same row and/or text elements in the same column in the original image.

[0073] Similar to constructing the first-type sample, a small quantity (for example, one to two thousand samples) of original images can be collected. First, the input data of the second-type sample is obtained based on the OCR output corresponding to the original image. Then, row and column information in the original image or the OCR output is labeled, to obtain the label data. The label data is elements in the same row and/or elements in the same column in the original image. A labeling rule is as follows: If vertical coordinates of two text elements are close to each other and a height error falls within a preset range, it indicates that the two text elements are in the same row; and if horizontal coordinates of the two text elements are close to each other and a width error falls within a preset range, it indicates that the two text elements are in the same column.

[0074] The input data and the label data are used to subsequently perform fine-tuning training on the pretrained large language model, so that the fine-tuned large language model completes a pre-processing task of recognizing the row and column information of the element in the OCR output before extracting the entity information. The second-type sample includes the input data and the label data in this example. In another example, a third-type sample in the training set is further constructed. Input data of the third-type sample is the above-mentioned OCR output, label data is a text element obtained after data cleaning is performed on the OCR output, and the data cleaning includes clearing noise and a scrambling code.

[0075] Similar to constructing the first-type sample, a small quantity (for example, one to two thousand samples) of original images can be collected. First, the input data of the third-type sample is obtained based on the OCR output corresponding to the original image. Then, data cleaning is

performed on the OCR output, to obtain label data. The data cleaning includes clearing noise and a scrambling code. The label data is a text element after data cleaning is performed on the OCR output.

[0076] For example, an example of an OCR output in the JSON format is still used. A text element that can be obtained: Transaction ID: 2023-XXXX-1232-1234◢◣-2231. It can be learned that due to a non-standard and irregular original image, the text element recognized by the OCR model includes the noise and the scrambling code "◢◣". The label data is obtained by performing a data cleaning operation on the OCR output is Transaction ID: 2023-XXXX-1232-1234-2231.

[0077] The input data and the label data are used to subsequently perform fine-tuning training on the pretrained large language model, so that the fine-tuned large language model completes a pre-processing task of data cleaning and verification on the OCR output before extracting entity information. The third-type sample includes the input data and the label data in this example.

[0078] Step S202: Perform fine-tuning on the pretrained large language model based on the training set.

[0079] In an embodiment, the training set includes the first-type sample, the second-type sample, and the third-type sample, and the pretrained large language model is fine-tuned based on the samples.

[0080] As described above, the second-type sample and the third-type sample are respectively used to train the large language model to recognize the row and column information of the text element and perform data cleaning on the text element, to facilitate subsequent extraction of the entity information. Therefore, tasks corresponding to the second-type sample and the third-type sample are optional in an entire fine-tuning process. The tasks corresponding to the two types of samples have no special requirement for execution time, and can be sequentially executed, or can be executed in a unified manner after a subset of the two types of samples is obtained through combination.

[0081] The first-type sample is used to train the large language model to extract the entity information. Execution time of the task is after those of the tasks corresponding to the second-type sample and the third-type sample, and the task is necessary in an entire fine-tuning process, and has a most complex execution process.

[0082] For example, execution processes of the three types of samples are arranged in the following sequence, to gradually perform fine-tuning training on the large language model from

easy to difficult, including:

[0083] First, first fine-tuning is performed on the large language model by using a subset of the second-type sample and the third-type sample.

[0084] In an implementation, when first fine-tuning is performed, one time of fine-tuning is first performed by using a subset of the second-type sample, and then another time of fine-tuning is performed based on the previous time of fine-tuning and a subset of the third-type sample.

[0085] In another implementation, when first fine-tuning is performed, one time of fine-tuning is performed on a subset of a dataset obtained by combining the second-type sample and the third-type sample. A sequence of the second-type sample and the third-type sample in the subset is disrupted.

[0086] When a corresponding subset is selected from the above-mentioned various sample sets, key features and categories of tasks corresponding to the various samples need to be covered. Then, the large language model is trained by using a selected sample subset. In this process, the large language model learns of features and patterns of the sample subset.

[0087] Then, after first fine-tuning, second fine-tuning is performed on the large language model by using the first-type sample.

[0088] When second fine-tuning is performed, one time of fine-tuning is performed by using the first-type sample.

[0089] First fine-tuning is performed, so that the large language model can adapt to the second-type sample and a preprocessing task corresponding to the second-type sample. Second fine-tuning is performed, so that the large language model can adapt to an entity information extraction task. Therefore, in an entire fine-tuning process, first fine-tuning is first performed on the large language model, to shorten time consumed for subsequent second fine-tuning, and improve efficiency of performing fine-tuning training on the large language model on the whole.

[0090] The following describes in detail, with reference to a specific example, a process of performing fine-tuning training on the pretrained large language model.

[0091] For example, FIG. 3 is a schematic diagram of fine-tuning of a large language model according to an example embodiment. The fine-tuning process is applicable to each time of fine-tuning in the first fine-tuning or the second fine-tuning.

[0092] To reduce resources and costs consumed by training, in this embodiment, the large language model is fine-tuned by using a bypass network.

[0093] For example, the bypass network is a low-rank adaptation (LORA) network.

[0094] Before fine-tuning training is performed, the training set that includes the first-type sample, the second-type sample, and the third-type sample is constructed in the method described in step S201. As described above, each type of sample has a corresponding task. First fine-tuning is performed by using the subset of the second-type sample and the third-type sample, and second fine-tuning is performed by using the first-type sample.

[0095] The pretrained large language model is selected as shown on a left side in FIG. 3, and has a pretrained weight pretrained on a large dataset. Although the pretrained weight performs well in an original task, the pretrained weight needs to be adjusted for the pretraining task and the entity information extraction task, to capture features related to these new tasks.

[0096] The large language model includes a plurality of network layers such as an input layer, a hidden layer, and an output layer. Each layer corresponds to a weight matrix W, and the pretrained weight provides an initial weight matrix for each layer, to help avoid training from zero to adapt to the new task, thereby accelerating a convergence speed. The weight matrix W can be a square matrix whose dimension is d, and can be represented as $W \in \mathbb{R}^{d \times d}$. A higher dimension d indicates a stronger processing capability of the large language model and a more complex processing task that can be completed. A quantity of parameters of the large language model is usually counted in billions, and these parameters are distributed in all weight matrices of the model.

[0097] The LORA bypass network designed for the model is shown on a right side in FIG. 3, is a small trainable neural network, and is connected to the large language model in parallel. The LORA bypass network is introduced for a specific part of the large language model, and the specific part relates to network layers in the large language model that are related to the preprocessing task or the entity information extraction task and weight matrices that are in these network layers and that are related to the preprocessing task or the entity information extraction task.

[0098] The LORA bypass network includes a plurality of low-rank matrices A, and a weight matrix related to the specific part is approximated through low-rank approximation. In other words, an original large matrix is approximated by using a product of several relatively small matrices, so that the LORA bypass network has fewer parameters than the large language model. A rank r of the low-rank matrix A is determined based on a computing resource that can be provided by a training platform and complexity of the training task.

[0099] Based on the above-mentioned structure, during fine-tuning training, the pretrained weight is kept unchanged, and only a parameter of the low-rank matrix is updated, so that the large language model adapts to the pre-processing task or the entity information extraction task.

[0100] Specifically, before fine-tuning training is performed, a parameter of the LORA bypass network is randomly initialized or initialized in another heuristic manner. For example, as shown in FIG. 3, random sampling is performed on a normal distribution whose mean value is 0 and variance is σ^2 to initialize the low-rank matrix A, which can be represented as $A = N(0, \sigma^2)$. A bias term of each neuron in the LORA bypass network is initialized to 0, and can be represented as B=0.

[0101] During fine-tuning training, input data corresponding to a sample subset related to first fine-tuning or a sample related to second fine-tuning is input into the large language model and the bypass network in batches based on a length d, and is jointly processed by the large language model and the bypass network, to obtain a predicted output y after summarization. A summarization manner h includes a weighted sum, a series connection, or a more complex fusion policy.

[0102] A difference Loss between the predicted output y and label data y' of the sample subset or the sample is determined by using a loss function. The parameter of the LORA bypass network is updated based on the difference Loss, so that the large language model adapts to the pre-processing task and the entity information extraction task.

[0103] Optionally, a small part of pretrained weights of the large language model can also be updated synchronously based on the difference Loss, so that the large language model adapts to the new task more quickly.

[0104] In the above-mentioned fine-tuning training process, the LoRA bypass network is introduced based on the pretrained weight of the large language model to perform parameter updating, so as to adapt to the new task. Because only a small quantity of trainable parameters are introduced into the LoRA bypass network, the fine-tuning process is more rapid and cost-effective. In addition, because the pretrained weight is possibly not adjusted or only a small part of the pretrained weights is possibly adjusted, a generalization capability of the large language model is maintained, and interference to pretraining knowledge is reduced.

[0105] In the above-mentioned process shown in FIG. 3, a prompt can also be constructed based on input data corresponding to any sample in the training set. The prompt is input into the

large language model and subsequent fine-tuning training is performed for a bypass network set for the large language model.

[0106] For example, constructing the prompt based on the input data corresponding to any sample in the training set includes: selecting one of a plurality of preset prompt templates as a current template, and filling input data corresponding to the sample in the current template, to obtain the prompt. Any one of the plurality of prompt templates includes a task description part, used to describe a background knowledge part of a task principle, and a to-be-filled task content part. Construction of the prompt is changed and adjusted based on a feature of the OCR output and a requirement of the pretrained large language model.

[0107] Optionally, construction of the prompt further includes: designing a valid prompt based on a preset output format of the large language model, and guiding the large language model to generate a proper output for the entity information extraction task.

[0108] The following is a prompt example:

//***Task description part***

"Please help us extract information such as an ID, an amount, a time, and a file. Understand problem logic and then make determining.\n"

"\n"+

"You can use the following information as knowledge to supplement determining:\n"+

"{knowledgebase}\n"+

"\n"+

"You need to use the extracted information in the following JSON format as an answer. When the information includes a plurality of values, a sequence number is added after a JSON KEY value for distinguishing:\n"+

"{JSON template}\n"+

//***To-be-filled task content part***

"The input information is as follows:\n"+

"{INPUT_OCR}\n"+

"Now give an answer. Do not give an explanation or reason:\n"+

[0109] In the above-mentioned prompt example, the task description part is used to describe a to-be-executed task, the background knowledge part (the knowledgebase part in the above-mentioned example) is used to describe a principle and background knowledge of the task, and the task content part is used to fill input data (namely, the OCR output) of a current sample. In addition, the prompt further includes a template structure in an output format.

[0110] Specifically, background knowledge in a target field that is used to supplement determining can be input into the knowledgebase part, and can be in a word format or a txt format.

[0111] The JSON template example is a template structure output by the large language model. For example, one JSON template is as follows: {"Date": "XXXX", "ID": "XXXX", "AMOUNT": "XXXX", "Date2": "XXXX"}, "Date", "ID", "AMOUNT", "Date", or "Date2" represents an entity information label that needs to be extracted, and "XXXX" represents the entity information.

[0112] INPUT_OCR in the task content part indicates that the input is an OCR output obtained by extracting original image information based on the OCR model, and can be in an XML format or a JSON format.

[0113] Therefore, when fine-tuning training is performed on the large language model by using the training set, the above-mentioned prompt template is also integrated into training data for fine-tuning together, to guide the large language model to output a correct answer, and further improve accuracy of an output of the large language model. In addition, during training, the prompt has a plurality of similar versions. To be specific, fine-tuning can be performed on the above-mentioned example prompt version, to enhance reasoning capability and generalization of the language model in this scenario.

[0114] It can be understood that, in a subsequent application process of the large language model, the prompt template can also be constructed and adjusted based on an actual application situation, so that relatively high accuracy of the output of the large language model is kept.

[0115] For example, FIG. 4 is a flowchart of an entity information extraction method according to an example embodiment. The method can be performed by using a computing platform, and the computing platform can be embodied as any computing unit, server, device, device cluster, etc. that has a computing and processing capability. The entity information extraction method includes the following steps.

[0116] Step S401: Obtain a fine-tuned large language model.

[0117] In this embodiment, a pretrained large language model can be fine-tuned in the method described in FIG. 2, so that the fine-tuned large language model adapts to the above-mentioned plurality of pre-processing tasks and the above-mentioned plurality of entity information extraction tasks.

[0118] Step S402: Perform text recognition on a target image by using an OCR model, to obtain a target OCR output. The target OCR output includes a recognized text element and location

information of each text element.

[0119] In this embodiment, the target image is a to-be-processed original image, and the target OCR output is data obtained by performing text recognition on the target image by using the OCR model.

[0120] Step S403: Construct a prompt based on the target OCR output, where the prompt indicates an entity recognition task.

[0121] In this embodiment, the prompt is constructed based on the target OCR in the method described in step S201. The prompt guides the fine-tuned pretrained large language model to execute the entity information recognition task.

[0122] Step S404: Input the prompt into the large language model, to obtain entity information output by the large language model.

[0123] In this embodiment, the prompt further includes: indicating an output format of the large language model, and inputting the prompt into the large language model, to obtain the entity information output by the large language model through recognition. An output format of the entity information meets a requirement of the prompt.

[0124] It should be noted that although operations of the method in the embodiments of this specification are described in a particular order in the above-mentioned embodiments, it is not required or implied that these operations need to be performed in the particular order or that all the shown operations need to be performed to achieve a desired result. In the contrary, an execution sequence of steps depicted in the flowchart can be changed. Additionally or alternatively, some steps can be omitted, a plurality of steps can be combined into one step for execution, and/or one step can be decomposed into a plurality of steps for execution.

[0125] The specification also provides a computer-readable non-transitory storage medium, where the storage medium stores a computer program, which, when executed by a processor, can be used to perform one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Herein, the computer-readable non-transitory storage medium or media may include one or more semiconductor-based or other integrated circuits (ICs) (such, as for example, field-programmable gate arrays (FPGAs) or application-specific ICs (ASICs)), hard disk drives (HDDs), hybrid hard drives (HHDs), optical discs, optical disc drives (ODDs), magneto-optical discs, magneto-optical drives, floppy diskettes, floppy disk drives (FDDs), magnetic tapes, solid-state drives (SSDs), RAM-drives, SECURE

DIGITAL cards or drives, any other suitable computer-readable non-transitory storage media, or any suitable combination of two or more of these, where appropriate. A computer-readable non-transitory storage medium may be volatile, non-volatile, or a combination of volatile and non-volatile, where appropriate.

[0126] The specification also provides a computer system. FIG. 5 illustrates an example computer system 1000. In particular embodiments, one or more computer systems 1000 perform one or more steps of one or more methods described or illustrated herein. In particular embodiments, one or more computer systems 1000 provide functionality described or illustrated herein. In particular embodiments, software running on one or more computer systems 1000 performs one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Particular embodiments include one or more portions of one or more computer systems 1000. Herein, reference to a computer system may encompass a computing device, and vice versa, where appropriate. Moreover, reference to a computer system may encompass one or more computer systems, where appropriate.

[0127] This disclosure contemplates any suitable number of computer systems 1000. This disclosure contemplates computer system 1000 taking any suitable physical form. As example and not by way of limitation, computer system 1000 may be an embedded computer system, a system-on-chip (SOC), a single-board computer system (SBC) (such as, for example, a computer-on-module (COM) or system-on-module (SOM)), a desktop computer system, a laptop or notebook computer system, an interactive kiosk, a mainframe, a mesh of computer systems, a mobile telephone, a personal digital assistant (PDA), a server, a tablet computer system, or a combination of two or more of these. Where appropriate, computer system 1000 may include one or more computer systems 1000; be unitary or distributed; span multiple locations; span multiple machines; span multiple data centers; or reside in a cloud, which may include one or more cloud components in one or more networks. Where appropriate, one or more computer systems 1000 may perform without substantial spatial or temporal limitation one or more steps of one or more methods described or illustrated herein. As an example and not by way of limitation, one or more computer systems 1000 may perform in real time or in batch mode one or more steps of one or more methods described or illustrated herein. One or more computer systems 1000 may perform at different times or at different locations one or more steps of one or more methods described or illustrated herein, where appropriate.

[0128] In particular embodiments, computer system 1000 includes a processor 1002, memory 1004, storage 1006, an input/output (I/O) interface 1008, a communication interface 1010, and a bus 1012. Although this disclosure describes and illustrates a particular computer system having a particular number of particular components in a particular arrangement, this disclosure contemplates any suitable computer system having any suitable number of any suitable components in any suitable arrangement.

[0129] In particular embodiments, processor 1002 includes hardware for executing instructions, such as those making up a computer program. As an example and not by way of limitation, to execute instructions, processor 1002 may retrieve (or fetch) the instructions from an internal register, an internal cache, memory 1004, or storage 1006; decode and execute them; and then write one or more results to an internal register, an internal cache, memory 1004, or storage 1006. In particular embodiments, processor 1002 may include one or more internal caches for data, instructions, or addresses. This disclosure contemplates processor 1002 including any suitable number of any suitable internal caches, where appropriate. As an example and not by way of limitation, processor 1002 may include one or more instruction caches, one or more data caches, and one or more translation lookaside buffers (TLB s). Instructions in the instruction caches may be copies of instructions in memory 1004 or storage 1006, and the instruction caches may speed up retrieval of those instructions by processor 1002. Data in the data caches may be copies of data of 80 in memory 1004 or storage 1006 for instructions executing at processor 1002 to operate on; the results of previous instructions executed at processor 1002 for access by subsequent instructions executing at processor 1002 or for writing to memory 1004 or storage 1006; or other suitable data. The data caches may speed up read or write operations by processor 1002. The TLBs may speed up virtual-address translation for processor 1002. In particular embodiments, processor 1002 may include one or more internal registers for data, instructions, or addresses. This disclosure contemplates processor 1002 including any suitable number of any suitable internal registers, where appropriate. Where appropriate, processor 1002 may include one or more arithmetic logic units (ALUs); be a multi-core processor; or include one or more processors 1002. Although this disclosure describes and illustrates a particular processor, this disclosure contemplates any suitable processor.

[0130] In particular embodiments, memory 1004 includes main memory for storing instructions for processor 1002 to execute or data for processor 1002 to operate on. As an example

and not by way of limitation, computer system 1000 may load instructions from storage 1006 or another source (such as, for example, another computer system 1000) to memory 1004. Processor 1002 may then load the instructions from memory 1004 to an internal register or internal cache. To execute the instructions, processor 1002 may retrieve the instructions from the internal register or internal cache and decode them. During or after execution of the instructions, processor 1002 may write one or more results (which may be intermediate or final results) to the internal register or internal cache. Processor 1002 may then write one or more of those results to memory 1004. In particular embodiments, processor 1002 executes only instructions in one or more internal registers or internal caches or in memory 1004 (as opposed to storage 1006 or elsewhere) and operates only on data in one or more internal registers or internal caches or in memory 1004 (as opposed to storage 1006 or elsewhere). One or more memory buses (which may each include an address bus and a data bus) may couple processor 1002 to memory 1004. Bus 1012 may include one or more memory buses, as described below. In particular embodiments, one or more memory management units (MMUs) reside between processor 1002 and memory 1004 and facilitate accesses to memory 1004 requested by processor 1002. In particular embodiments, memory 1004 includes random access memory (RAM). This RAM may be volatile memory, where appropriate. Where appropriate, this RAM may be dynamic RAM (DRAM) or static RAM (SRAM). Moreover, where appropriate, this RAM may be single-ported or multi-ported RAM. This disclosure contemplates any suitable RAM. Memory 1004 may include one or more memories 1004, where appropriate. Although this disclosure describes and illustrates particular memory, this disclosure contemplates any suitable memory.

[0131] In particular embodiments, storage 1006 includes mass storage for data or instructions. As an example and not by way of limitation, storage 1006 may include a hard disk drive (HDD), a floppy disk drive, flash memory, an optical disc, a magneto-optical disc, magnetic tape, or a Universal Serial Bus (USB) drive or a combination of two or more of these. Storage 1006 may include removable or non-removable (or fixed) media, where appropriate. Storage 1006 may be internal or external to computer system 1000, where appropriate. In particular embodiments, storage 1006 is non-volatile, solid-state memory. In particular embodiments, storage 1006 includes read-only memory (ROM). Where appropriate, this ROM may be mask-programmed ROM, programmable ROM (PROM), erasable PROM (EPROM), electrically erasable PROM (EEPROM), electrically alterable ROM (EAROM), or flash memory or a combination of two or

more of these. This disclosure contemplates mass storage 1006 taking any suitable physical form. Storage 1006 may include one or more storage control units facilitating communication between processor 1002 and storage 1006, where appropriate. Where appropriate, storage 1006 may include one or more storages 1006. Although this disclosure describes and illustrates particular storage, this disclosure contemplates any suitable storage.

[0132] In particular embodiments, I/O interface 1008 includes hardware, software, or both, providing one or more interfaces for communication between computer system 1000 and one or more I/O devices. Computer system 1000 may include one or more of these I/O devices, where appropriate. One or more of these I/O devices may enable communication between a person and computer system 1000. As an example and not by way of limitation, an I/O device may include a keyboard, keypad, microphone, monitor, mouse, printer, scanner, speaker, still camera, stylus, tablet, touch screen, trackball, video camera, another suitable I/O device or a combination of two or more of these. An I/O device may include one or more sensors. This disclosure contemplates any suitable I/O devices and any suitable I/O interfaces 1008 for them. Where appropriate, I/O interface 1008 may include one or more device or software drivers enabling processor 1002 to drive one or more of these I/O devices. I/O interface 1008 may include one or more I/O interfaces 1008, where appropriate. Although this disclosure describes and illustrates a particular I/O interface, this disclosure contemplates any suitable I/O interface.

[0133] In particular embodiments, communication interface 1010 includes hardware, software, or both providing one or more interfaces for communication (such as, for example, packet-based communication) between computer system 1000 and one or more other computer systems 1000 or one or more networks. As an example and not by way of limitation, communication interface 1010 may include a network interface controller (NIC) or network adapter for communicating with an Ethernet or other wire-based network or a wireless NIC (WNIC) or wireless adapter for communicating with a wireless network, such as a WI-FI network. This disclosure contemplates any suitable network and any suitable communication interface 1010 for it. As an example and not by way of limitation, computer system 1000 may communicate with an ad hoc network, a personal area network (PAN), a local area network (LAN), a wide area network (WAN), a metropolitan area network (MAN), or one or more portions of the Internet or a combination of two or more of these. One or more portions of one or more of these networks may be wired or wireless. As an example, computer system 1000 may communicate with a wireless PAN (WPAN) (such as, for example, a

BLUETOOTH WPAN), a WI-FI network, a WI-MAX network, a cellular telephone network (such as, for example, a Global System for Mobile Communications (GSM) network), or other suitable wireless network or a combination of two or more of these. Computer system 1000 may include any suitable communication interface 1010 for any of these networks, where appropriate. Communication interface 1010 may include one or more communication interfaces 1010, where appropriate. Although this disclosure describes and illustrates a particular communication interface, this disclosure contemplates any suitable communication interface.

[0134] In particular embodiments, bus 1012 includes hardware, software, or both coupling components of computer system 1000 to each other. As an example and not by way of limitation, bus 1012 may include an Accelerated Graphics Port (AGP) or other graphics bus, an Enhanced Industry Standard Architecture (EISA) bus, a front-side bus (FSB), a HYPERTRANSPORT (HT) interconnect, an Industry Standard Architecture (ISA) bus, an INFINIBAND interconnect, a low-pin-count (LPC) bus, a memory bus, a Micro Channel Architecture (MCA) bus, a Peripheral Component Interconnect (PCI) bus, a PCI-Express (PCIe) bus, a serial advanced technology attachment (SATA) bus, a Video Electronics Standards Association local (VLB) bus, or another suitable bus or a combination of two or more of these. Bus 1012 may include one or more buses 1012, where appropriate. Although this disclosure describes and illustrates a particular bus, this disclosure contemplates any suitable bus or interconnect.

[0135] It should be further noted that, the terms "include", "comprise", or any other variant thereof are intended to cover a non-exclusive inclusion, so that a process, a method, a product or a device that includes a list of elements not only includes those elements but also includes other elements which are not expressly listed, or further includes elements inherent to such process, method, product or device. Without more constraints, an element preceded by "includes a …" does not preclude the existence of additional identical elements in the process, method, product or device that includes the element.

[0136] The embodiments of this specification are described in a progressive way. For same or similar parts in the embodiments, references can be made to each other. Each embodiment focuses on a difference from another embodiment. Particularly, some system embodiments are briefly described because they are basically similar to some method embodiments. For related parts, references can be made to related descriptions in some method embodiments.

[0137] The foregoing descriptions are merely embodiments of this specification, and are not

intended to limit this specification. A person skilled in the art can make various changes and changes to this specification. Any modification, equivalent replacement, improvement, etc. made without departing from the spirit and principle of this specification shall fall within the scope of the claims in this application.

CLAIMS

What is claimed is:

1. A large model training method, comprising:

constructing a training set, wherein the training set comprises at least a first-type sample, input data of the first-type sample comprises an OCR output obtained by performing text recognition on an original image by using an OCR model, the OCR output comprises a recognized text element and location information of each element, and label data of the first-type sample comprises entity information in the original image; and

performing fine-tuning on a pretrained large language model based on the training set.

2. The method according to claim 1, wherein the training set further comprises a second-type sample, input data of the second-type sample is the OCR output, and label data of the second-type sample is text elements in a same row and/or text elements in a same column in the original image.

3. The method according to claim 2, wherein the training set further comprises a third-type sample, input data of the third-type sample is the OCR output, label data of the third-type sample is a text element after data cleaning is performed on the OCR output, and the data cleaning comprises: clearing noise and a scrambling code.

4. The method according to claim 3, wherein the performing fine-tuning on a pretrained large language model based on the training set comprises:

performing first fine-tuning on the large language model by using a subset of the second-type sample and the third-type sample; and

after first fine-tuning, performing second fine-tuning on the large language model by using the first-type sample.

5. The method according to claim 4, wherein a sequence of the second-type sample and the third-type sample in the subset is disrupted.

6. The method according to claim 1, wherein the performing fine-tuning on a pretrained large language model based on the training set comprises:

constructing a prompt based on input data corresponding to any sample in the training set;

inputting the prompt into the large language model and a bypass network set for the large language model, and obtaining a predicted output by performing joint processing by the large language model and the bypass network; and

adjusting a parameter of the bypass network based on the predicted output and the label data of the above-mentioned sample.

7. The method according to claim 1, wherein the bypass network is a LoRA network.

8. The method according to claim 6, wherein the constructing a prompt based on input data corresponding to any sample in the training set comprises:

selecting one of a plurality of preset prompt templates as a current template, and filling the input data corresponding to the sample in the current template, to obtain the prompt, wherein any one of the plurality of prompt templates comprises a task description part, used to describe a background knowledge part of a task principle, and a to-be-filled task content part.

9. The method according to claim 1, wherein the original image is an image in a work order, a form, or an order.

10. An entity information extraction method, comprising:

obtaining a fine-tuned large language model according to any one of claims 1 to 9;

performing text recognition on a target image by using an OCR model, to obtain a target OCR output, wherein the target OCR output comprises a recognized text element and location information of each text element;

constructing a prompt based on the target OCR output, wherein the prompt indicates an entity recognition task; and

inputting the prompt into the large language model, to obtain entity information output by the large language model.

11. A computer system, comprising a storage, a processor, and a computer program that is stored in the storage and that is capable of running on the processor, wherein the processor performs the following target operation for at least one time when executing the program, and implements the following steps by performing the target operation:

constructing a training set, wherein the training set comprises at least a first-type sample, input data of the first-type sample comprises an OCR output obtained by performing text recognition on an original image by using an OCR model, the OCR output comprises a recognized text element and location information of each element, and label data of the first-type sample comprises entity information in the original image; and

performing fine-tuning on a pretrained large language model based on the training set.

12. The computer system according to claim 11, wherein the training set further comprises a

second-type sample, input data of the second-type sample is the OCR output, and label data of the second-type sample is text elements in a same row and/or text elements in a same column in the original image.

13. The computer system according to claim 12, wherein the training set further comprises a third-type sample, input data of the third-type sample is the OCR output, label data of the third-type sample is a text element after data cleaning is performed on the OCR output, and the data cleaning comprises: clearing noise and a scrambling code.

14. The computer system according to claim 13, wherein the performing fine-tuning on a pretrained large language model based on the training set comprises:

performing first fine-tuning on the large language model by using a subset of the second-type sample and the third-type sample; and

after first fine-tuning, performing second fine-tuning on the large language model by using the first-type sample.

15. The computer system according to claim 14, wherein a sequence of the second-type sample and the third-type sample in the subset is disrupted.

16. The computer system according to claim 11, wherein the performing fine-tuning on a pretrained large language model based on the training set comprises:

constructing a prompt based on input data corresponding to any sample in the training set;

inputting the prompt into the large language model and a bypass network set for the large language model, and obtaining a predicted output by performing joint processing by the large language model and the bypass network; and

adjusting a parameter of the bypass network based on the predicted output and the label data of the above-mentioned sample.

17. The computer system according to claim 11, wherein the bypass network is a LoRA network.

18. The computer system according to claim 16, wherein the constructing a prompt based on input data corresponding to any sample in the training set comprises:

selecting one of a plurality of preset prompt templates as a current template, and filling the input data corresponding to the sample in the current template, to obtain the prompt, wherein any one of the plurality of prompt templates comprises a task description part, used to describe a background knowledge part of a task principle, and a to-be-filled task content part.

19. A computer system, comprising a storage, a processor, and a computer program that is stored in the storage and that is capable of running on the processor, wherein the processor performs the following target operation for at least one time when executing the program, and implements the following steps by performing the target operation:

obtaining a fine-tuned large language model according to any one of claims 1 to 9;

performing text recognition on a target image by using an OCR model, to obtain a target OCR output, wherein the target OCR output comprises a recognized text element and location information of each text element;

constructing a prompt based on the target OCR output, wherein the prompt indicates an entity recognition task; and

inputting the prompt into the large language model, to obtain entity information output by the large language model.

20. A computer-readable storage medium, wherein the computer-readable storage medium stores a computer program, and when the computer program is executed on a computer, the computer is enabled to perform the method according to any one claims 1 to 9 or claim 10.