APPLICATION PROGRAM INSPECTION METHOD AND APPARATUS, STORAGE MEDIUM, AND COMPUTER SYSTEM

TECHNICAL FIELD

[0001] One or more embodiments of this specification relate to the computer field, and in particular, to an application program inspection method and apparatus based on reinforcement learning, a storage medium, and a computer system.

BACKGROUND

[0002] When an application program that can run on a terminal is inspected, a large quantity of manual interventions usually needs to be performed. For example, test personnel needs to identify a trigger area that can be used to perform tap interaction on a page displayed by the terminal for the application program, and performs a tap action on the trigger area to determine, based on an execution result of the tap action, whether a bug exists on the current page.

[0003] It is hoped that there is a new technical solution to enable more efficiently and comprehensively inspect an application program.

SUMMARY

[0004] One or more embodiments of this specification provide an application program inspection method and apparatus based on reinforcement learning, a storage medium, and a computer system.

[0005] This specification provides an application program inspection method based on reinforcement learning. The method includes: determining, from a first page displayed by a terminal for a target application program, N trigger areas that are used to perform tap interaction; determining a page embedding of the first page based on the N trigger areas; inputting the page embedding to a reinforcement learning model, so that the reinforcement learning model outputs action decision information corresponding to action space, where the action space includes several preset actions and N trigger actions that are respectively used to trigger the N trigger areas; performing a target action on the first page based on the action decision information, so that the terminal displays a second page; and obtaining an environment reward for performing the target action on the first page, where a value taken by the environment reward when the second page is

a newly appeared page on the terminal is greater than a value taken by the environment reward when the second page is not a newly appeared page on the terminal, and the environment reward is used to update the reinforcement learning model.

[0006] This specification provides a computer system, 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 executes the computer program to implement the following steps: determining, from a first page displayed by a terminal for a target application program, N trigger areas that are used to perform tap interaction; determining a page embedding of the first page based on the N trigger areas; inputting the page embedding to a reinforcement learning model, so that the reinforcement learning model outputs action decision information corresponding to action space, where the action space includes several preset actions and N trigger actions that are respectively used to trigger the N trigger areas; performing a target action on the first page based on the action decision information, so that the terminal displays a second page; and obtaining an environment reward for performing the target action on the first page, where a value taken by the environment reward when the second page is a newly appeared page on the terminal is greater than a value taken by the environment reward when the second page is not a newly appeared page on the terminal, and the environment reward is used to update the reinforcement learning model.

[0007] This specification provides an application program inspection apparatus based on reinforcement learning, including: a target detection unit, configured to determine, from a first page displayed by a terminal for a target application program, N trigger areas that are used to perform tap interaction; an embedding processing unit, configured to determine a page embedding of the first page based on the N trigger areas; a model invoking unit, configured to input the page embedding to a reinforcement learning model, so that the reinforcement learning model outputs action decision information corresponding to action space, where the action space includes several preset actions and N trigger actions that are respectively used to trigger the N trigger areas; an action execution unit, configured to perform a target action on the first page based on the action decision information, so that the terminal displays a second page; and a reward calculation unit, configured to obtain an environment reward for performing the target action on the first page, where a value taken by the environment reward when the second page is a newly appeared page on the terminal is greater than a value taken by the environment reward when the second page is not a newly appeared page on the terminal, and the environment reward is used to update the reinforcement learning model.

[0008] This specification provides a computer-readable nonvolatile storage medium. The storage medium stores a computer program, and the computer program is executed by a processor to implement the foregoing application program inspection method based on reinforcement

learning.

[0009] The technical solutions used in embodiments of this specification can achieve at least one of the following beneficial effects: N trigger areas that can be used to perform tap interaction on a page displayed by a terminal for a target application program at a current moment can be automatically identified, and a reinforcement learning model is used to provide better action decision information for the page displayed by the terminal for the target application program at the current moment, so that after a target action is performed on the page based on the action decision information in a subsequent process, the terminal can display, as much as possible, a page that has not been displayed for the target application program, and the terminal can explore and display, as much as possible, all pages provided by the target application program, to reduce, as much as possible, a frequency of repeatedly inspecting a same page provided by the target application program, thereby helping inspect the target application program more quickly, efficiently, and comprehensively.

BRIEF DESCRIPTION OF DRAWINGS

[0010] To describe technical solutions in embodiments of this specification more clearly, the following briefly describes accompanying drawings required for describing embodiments. Clearly, the accompanying drawings in the following description show only some embodiments of this specification, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0011] FIG. 1 is a schematic diagram of an example of a reinforcement learning system according to an embodiment of this specification;

[0012] FIG. 2 is a diagram of a system architecture of a technical solution according to an embodiment of this specification;

[0013] FIG. 3 is a flowchart of an application program inspection method based on reinforcement learning according to an embodiment of this specification;

[0014] FIG. 4 is a schematic diagram of an application program inspection apparatus based on reinforcement learning according to an embodiment of this specification; and

[0015] FIG. 5 is a schematic diagram of a structure of a computer system according to an embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

[0016] To make the objectives, technical solutions, and advantages of this specification clearer, the following clearly and comprehensively describes the technical solutions of this specification

with reference to specific embodiments and accompanying drawings of this specification. Clearly, the described embodiments are merely some but not all of 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.

[0017] A single application program usually provides a large quantity of pages that can be displayed on a terminal. The foregoing application program can be a mobile application program that can directly run on the terminal, or can be a small program that run relying on another mobile application program. A single page provided by an application program can usually include a plurality of elements, and the foregoing elements can usually include a character, an image, or a combination thereof. When a page is displayed on the terminal, a user can perform a trigger operation (for example, a tap operation) on an element on the displayed page, so that the terminal displays another page associated with the triggered element, thereby completing a transaction that the user expects to complete.

[0018] When the triggered element is not correctly associated with another page expected to be associated with, the terminal may continue to display the current page or directly terminate running a related application program. As a result, the transaction that the user expects to complete cannot be completed. Therefore, an application program needs to be inspected, to find an element that is on a large amount of page content provided by the application program and that is not correctly associated with another page.

[0019] As described above, when an application program is inspected through manual intervention, test personnel needs to identify and tap a trigger area that can be used to perform tap interaction on a page provided by the application program (that is, an element that can be used to perform tap interaction on the page provided by the application program). As a result, many problems exist, including the following examples: A speed of identifying and tapping the trigger area is relatively small, making it difficult to quickly inspect a large quantity of pages provided by the application program. Due to influence of subjective factors of the test personnel, some trigger areas on the page provided by the application program may be omitted and not correctly tapped, resulting in failure to inspect some pages provided by the application program. It is highly likely to repeatedly inspect a same page provided by the application program.

[0020] In view of this, embodiments of this specification provide at least an application program inspection method and apparatus based on reinforcement learning, a storage medium, and a computer system. N trigger areas that can be used to perform tap interaction on a page displayed by a terminal for a target application program at a current moment can be automatically identified, and a reinforcement learning model is used to provide better action decision information for the page, so that after a target action is performed on the page based on the action decision information

in a subsequent process, the terminal can display, as much as possible, a page that has not been displayed for the target application program, and the terminal can explore and display, as much as possible, all pages provided by the target application program, to reduce, as much as possible, a frequency of repeatedly inspecting a same page provided by the target application program, thereby helping inspect the target application program more quickly, efficiently, and comprehensively.

[0021] Reinforcement learning is an important field in machine learning, and emphasizes how to act based on an environment to maximize expected benefits. Reinforcement learning is a third basic machine learning method in addition to supervised learning and unsupervised learning. Unlike supervised learning, reinforcement learning does not require tagged input-output pairs, and focuses on balancing search and exploration of an unknown field and exploitation of existing knowledge.

[0022] FIG. 1 is a schematic diagram of an example of a reinforcement learning system according to an embodiment of this specification. As shown in FIG. 1, the reinforcement learning system can include an agent and an execution environment. The agent performs interaction and feedback with the execution environment to continuously perform learning, to optimize a policy of the agent. Specifically, the agent can observe and obtain a state of the execution environment, and determine, based on a specific policy, action decision information that needs to be provided for the state of the execution environment. When the action selected based on the action decision information is applied to the execution environment, the state of the execution environment is changed, and a feedback is generated to the agent. The feedback is also referred to as an environment reward. Then, the agent can determine, based on the obtained environment reward, whether the previously provided action decision information is correct, and whether a policy needs to be adjusted, to update a policy of the agent. In summary, the agent repeatedly and continuously observes the state, determines required action decision information, and receives a feedback, so that the agent can continuously update the policy of the agent. An ultimate goal is to learn a policy to maximize accumulated obtained environmental rewards. After the policy of the agent is fully updated, the agent can provide better action decision information in different states.

[0023] The foregoing policy is mapping from the state of the execution environment to the action decision information. The action decision information can include a deterministic action, or can include execution probabilities respectively corresponding to a plurality of actions included in action space.

[0024] FIG. 2 is a flowchart of an application program inspection method based on reinforcement learning according to an embodiment of this specification. The method describes an example of a process of inspecting a target application program. As described above, the target

application program can be a mobile application program that can run independently in a terminal, or the target application program can be a small program that run relying on a mobile application program. The terminal can include but is not limited to a personal computer, a mobile phone, a tablet computer, etc.

[0025] The method can be performed by any apparatus, device, platform, or device cluster that has a computing/processing capability. A terminal that runs the target application program corresponds to the execution environment in the reinforcement learning system. An entity configured to perform the method can be integrated into the execution environment, or can be integrated into the agent including a reinforcement learning model as a component of the agent, or can be divided into a plurality of modules, where some modules are integrated into the execution environment, and remaining modules are integrated into the agent.

[0026] As shown in FIG. 2, the method can include but is not limited to some or all of the following step S201 to step S215.

[0027] Step S201: Determine, from a first page displayed by the terminal for the target application program, N trigger areas that are used to perform tap interaction.

[0028] When a target application program needs to be inspected, test personnel can provide a ULR scheme corresponding to the target application program to the terminal, so that the terminal runs the target application program and displays a page provided by the target application program. As shown in FIG. 3, the first page refers to a current page that is actually presented by the terminal to a user at a current moment in a process of inspecting the target application program by using the terminal. In a process in which the terminal displays any page provided by the target application program, for example, a page P, a current page (that is, the first page) that is actually presented by the terminal to the user by using a corresponding display component usually includes not only the page P, but also may include a framework element provided by the terminal, for example, a battery level indication icon of a battery built in the terminal, time information, and a state indication icon corresponding to a network accessed by the terminal, where the framework element usually does not support tap interaction. Therefore, it is not difficult to understand that the N trigger areas used to perform tap interaction in step S201 essentially correspond to N elements included in a page provided by the target application program.

[0029] In a possible implementation, the first page can be processed by using a pre-trained target detection model, to mark the N trigger areas that can be used to perform tap interaction from the first page. Generally, the target detection model used to process the first page can be obtained by fine tuning a pre-training model. In a relatively typical example, a state-of-the-art (SOTA) model DINO may be used as a pre-training model, and a corresponding training data set is selected to fine-tune the DINO. A goal lies in the following: When a fine-tuned DINO is used to process a

page provided by the terminal for any application program, all trigger areas that can be used to perform tap interaction on the page can be well identified and marked.

[0030] In another possible implementation, page dump information of the first page can be obtained by using a corresponding page analysis tool. For example, related information of the first page is dumped into a dump.xml file from a memory of the terminal, and the dump.xml file includes the page dump information of the first page. The page dump information includes position information that is on the first page and that respectively corresponds to N elements included in a page provided by the target application program. Therefore, the N trigger areas that are used to perform tap interaction can be marked from the first page based on the position information that is on the first page and that respectively corresponds to the N elements.

[0031] It should be noted that a value of N is a specific quantity of trigger areas determined from the first page instead of a preset value.

[0032] Step S203: Determine a page embedding of the first page based on the N trigger areas.

[0033] Image areas on the first page that respectively correspond to the N trigger areas can be encoded to obtain image embeddings respectively corresponding to the N trigger areas. For example, the image areas on the first page that respectively correspond to the N trigger areas can be processed by using an encoder for contrastive language-image pre-training (CPID), to obtain the image embeddings respectively corresponding to the N trigger areas. Similarly, the position information that is on the first page and that respectively corresponding to the N trigger areas can be encoded to obtain position embeddings (denoted as Position embeddings) respectively corresponding to the N trigger areas. Further, image embeddings and Position embeddings that respectively correspond to the N trigger can be combined to obtain the page embedding of the first page.

[0034] An image area corresponding to a single trigger area on the first page is usually rectangular. Correspondingly, position information corresponding to the single trigger area on the first page can be expressed by using a horizontal coordinate minimum (x-min), a vertical coordinate minimum (y-min), a horizontal coordinate maximum (x-max), and a vertical coordinate maximum (y-max) that are of the image area corresponding to the trigger area in an image coordinate system. Generally, values of x-min, y-min, x-max, and y-max can be relatively large, and therefore can be first normalized to map the values of x-min, y-min, x-max, and y-max to 0 to 1. Then, normalized x-min, y-min, x-max, and y-max are encoded to obtain a corresponding Position embedding.

[0035] Although an example of a process of obtaining the image embeddings and the Position embeddings that respectively correspond to the N trigger areas is described above, it can be understood that the foregoing process is merely an example. For example, another encoder can be

further used to process the image area corresponding to the trigger area on the first page, and it is only required to ensure that an obtained image embedding and Position embedding can accurately express a corresponding image area and position information.

[0036] The foregoing image embedding and Position embedding usually can be feature vectors with a same dimension. On this basis, an image embedding and a Position embedding that correspond to a same trigger area can be first combined (for example, added or successively concatenated) to obtain embedding features respectively corresponding to the N trigger areas. Further, the page embedding of the first page can be formed by using the embedding features respectively corresponding to the N trigger areas.

[0037] Step S205: Input the page embedding of the first page to a reinforcement learning model, so that the reinforcement learning model outputs action decision information corresponding to action space, where the action space includes several preset actions and N trigger actions that are respectively used to trigger the N trigger areas.

[0038] As shown in FIG. 3, an entity configured to implement the method, for example, an application inspection apparatus based on reinforcement learning, can invoke the reinforcement learning model to process a current page displayed by the terminal, to obtain action decision information of the current page.

[0039] The reinforcement learning model belongs to the agent in the reinforcement learning system, and the page embedding of the first page can be understood as a state of the execution environment observed by the agent. A policy used in the reinforcement learning model can include a typical deep deterministic policy gradient (DDPG) algorithm, a soft actor-critic (SAC) algorithm, a Q-learning algorithm, a deep Q-network (DQN), etc. The SAC algorithm introduces information entropy as a part of a reward based on the DDPG algorithm, which can improve exploration efficiency and support efficient and stable learning in a relatively complex execution environment. Therefore, in this embodiment of this specification, the policy used in the reinforcement learning model is preferably the SAC algorithm.

[0040] Because trigger actions that may be performed on the first page are discrete actions, action space of the reinforcement learning model can be designed as distribution based on classification. In addition, some trigger actions supported by the first page, for example, sliding a screen upward, sliding a screen downward, and sliding a screen leftward, may not depend on the N trigger areas included on the first page. Sliding a screen upward and sliding a screen downward usually causes the terminal to display another page that is not completely the same as the first page, and sliding a screen leftward usually causes the terminal to return from the first page to a previous page that has been displayed. On this basis, it is assumed that any page displayed by the terminal for the target application program displays a maximum of M trigger areas, and it is assumed that

any page displayed by the terminal for the target application program further supports K trigger actions that do not depend on the trigger areas. In this case, the action space can include M trigger actions that are used to trigger the M trigger areas and K preset actions (trigger actions that do not depend on the trigger areas).

[0041] In a possible implementation, the first M trigger actions in the action space correspond to M trigger areas that may exist, and the last K trigger actions in the action space correspond to the foregoing K preset actions. Correspondingly, the N trigger areas can be sorted based on the position information that is on the first page and that respectively corresponds to the N trigger areas. For example, with reference to x-min included in the position information of the trigger area in the foregoing example, the N trigger areas are sorted in descending order of x-min respectively corresponding to the N trigger areas. If values of x-min respectively corresponding to at least two trigger areas are the same, the at least two trigger areas are sorted in descending order of y-min. Therefore, any i^{th} trigger area in the sorted N trigger areas corresponds to an i^{th} trigger action in the action space.

[0042] When processing the page embedding of the first page by using a corresponding policy, the reinforcement learning model can calculate execution probabilities respectively corresponding to the (M+K) trigger actions in the action space. Correspondingly, the action decision information output by the reinforcement learning model can be used to indicate the execution probabilities respectively corresponding to the (M+K) trigger actions in the action space. Alternatively, the reinforcement learning model can further select, from the (M+K) trigger actions, a trigger action with a relatively large execution probability as a target action expected to be executed, and the action decision information output by the reinforcement learning model can be used to indicate the target action.

[0043] In a more specific example, for example, the action space specifically includes (M+K) action sequence numbers such as 1 to M+K, where the first N action sequence numbers are in a one-to-one correspondence with the N trigger areas arranged in sequence, and the last K action sequence numbers are in a one-to-one correspondence with the K preset actions arranged in sequence. The reinforcement learning model can output, for the page embedding of the first page, execution probabilities respectively corresponding to the (M+K) action sequence numbers, or select a target action sequence number from the (M+K) action sequence numbers, and output the target action sequence number.

[0044] Step S207: Perform a target action on the first page based on the action decision information, so that the terminal displays a second page.

[0045] When the action decision information indicates the execution probabilities respectively corresponding to the (M+K) trigger actions in the action space, a trigger action with a relatively

large execution probability can be selected from the (M+K) actions as the target action.

[0046] When the target action belongs to the preset actions that do not depend on the trigger areas, the target action can be directly performed.

[0047] When the target action belongs to the N trigger actions corresponding to the N trigger actions, the target action can be performed based on position information that is on the first page and that is of a target trigger area corresponding to the target action, that is, an image area corresponding to the target trigger area on the first page is triggered based on the position information that is on the first page and that is of the target trigger area corresponding to the target action.

[0048] In a process of inspecting the target application program, the terminal needs to be able to access all pages provided by the target application program as comprehensively as possible, and be able to avoid repeatedly accessing a same page as much as possible, so that the application program is inspected more quickly, comprehensively, and efficiently. To achieve this objective, it is expected that after the target action is performed on the first page, the second page displayed by the terminal is a newly appeared page rather than a page that has been appeared.

[0049] To determine whether the second page is a newly appeared page, in a possible implementation, in step S209, a page code used to represent the second page is obtained, and whether the second page is a newly appeared page is determined based on the page code.

[0050] In a possible implementation, after the target application program starts to be inspected, a code set can be maintained. The code set includes a page code corresponding to a preset reference page and page code respectively corresponding to all pages displayed by the terminal for the target application program. The reference page refers to a last page displayed by the terminal before the target application program starts to be inspected, for example, before the test personnel triggers the terminal to start to run the target application program.

[0051] Correspondingly, page dump information of the second page can be obtained by using a corresponding page analysis tool, where the page dump information includes framework information of the second page, and the framework information includes position information that is on the second page and that respectively corresponds to a plurality of elements included on the second page. The plurality of elements include not only an element included on a page provided by the target application program, but also may include a framework element provided by the terminal. Position information of the framework element may adversely affect page similarity determining. Therefore, the position information of the framework element provided by the terminal can be excluded from the framework information. After the position information of the framework element is excluded, the framework information may still include position information of a large quantity of elements provided by the target application program. To help more efficiently

determine similarity, position information corresponding to an element that does not carry a character can continue to be excluded, and only position information that is on the first page and that respectively corresponds to at least one target element that is provided by the target application program and that carries a character is reserved. Further, the page code corresponding to the second page can be determined based on the position information respectively corresponding to the at least one target element. For example, the position information respectively corresponding to the at least one target element can be concatenated in a certain order, and an excessively long concatenation result can be clipped to pad an excessively short concatenation result, to obtain a page code with a predetermined length.

[0052] After the page code of the second page is obtained, whether the code set includes the page code of the second page can continue to be queried. If yes, it is determined that the second page is not a newly appeared page.

[0053] The foregoing process of determining whether the second page is a newly appeared page is an example. For example, a similarity between the second page and each of the preset reference page and pages that have been displayed by the terminal for the target application program can be directly calculated. When at least one similarity is greater than a similarity threshold, it is determined that the second page is not a newly appeared page.

[0054] Step S211: Obtain an environment reward for performing the target action on the first page, where a value taken by the environment reward when the second page is a newly appeared page on the terminal is greater than a value taken by the environment reward when the second page is not a newly appeared page on the terminal, and the environment reward is used to update the reinforcement learning model.

[0055] With reference to a training and working principle of the reinforcement learning model, the value taken by the environment reward when the second page is a newly appeared page on the terminal is set to be greater than the value taken by the environment reward when the second page is not a newly appeared page on the terminal, so that after the reinforcement learning model is updated by using the environment reward in a subsequent process, an updated reinforcement learning model can provide better action decision information for a page displayed by the terminal, and after the target action is subsequently performed based on corresponding action decision information on the page displayed by the terminal, the terminal can display, as much as possible, a page that is provided by the target application and that has not been displayed.

[0056] With reference to the foregoing description, some elements on a page provided by the target application program may not be correctly associated with other pages that expect to be associated with. When an element corresponding to the target trigger area is not correctly associated with another page that expects to be associated with, after the trigger action is performed

on the target trigger area on the first page, the terminal may continue to display the first page or terminate running of the target application program. If the terminal terminates running of the target application program, the terminal may display the foregoing reference page.

[0057] When the element corresponding to the target trigger area is not correctly associated with an another page that expects to be associated with, it means that the terminal cannot reach a new page by using the target trigger area and continue to inspect the target application program. Therefore, after the reinforcement learning model is updated by using the environment reward in the subsequent process, it is expected that the updated reinforcement learning model can provide better action decision information for a page displayed by the terminal, to avoid, as much as possible, continuing to perform the trigger action on the target trigger area. Based on this, when the second page is similar to the first page or the preset reference page that appears on the terminal, a value of the environment reward is a first preset value; or when the second page is similar to a page that appears on the terminal other than the first page and the preset reference page, a value of the environment reward is a second preset value, where the first preset value is less than the second preset value.

[0058] In a more specific example, when the second page is a newly appeared page, a value of the environment reward can be, for example, a preset value 1; or when the second page is not a newly appeared page, if the second page is similar to the first page or the preset reference page, a value of the environment reward can be, for example, a preset value −1; otherwise, a value of the environment reward can be, for example, 0.

[0059] In the process of inspecting the target application program, to enable, as much as possible based on the action decision information output by the reinforcement learning model, the terminal to display a page that has not been displayed for the target application program, optionally, in step S213, an exploration reward for performing the target action on the first page is obtained, where the exploration reward is used to update the reinforcement learning model.

[0060] As an intrinsic reward of the reinforcement learning model, the exploration reward can be obtained by processing the page embedding of the second page by using a random network distillation (RND) algorithm or an intrinsic curiosity module (ICM) algorithm. The RND algorithm is used as an example. As shown in FIG. 3, the RND algorithm includes a fixed target network that is randomly initialized and a predictor network that can be trained. An entity configured to implement the method, for example, the application inspection apparatus based on reinforcement learning, can invoke the RND algorithm to process a current page (that is, the second page) displayed by the terminal, to obtain processing results that are respectively output by the target network and the predictor network for the second page, and then can use a cosine distance between the two processing results as the exploration reward.

[0061] Referring to the foregoing description, when the second page is similar to the first page or the preset reference page that appears on the terminal, it indicates that there is a bug on a target page that is provided by the target application program and that is included on the first page, for example, the element corresponding to the target trigger area on the target page may be not correctly associate with another page that expects to be associated with. Based on this, to remind the test personnel that a bug exists on the target page, the following step S215 can be further performed: When the second page is similar to the first page or the preset reference page that appears on the terminal, error prompt information corresponding to the first page is generated.

[0062] For example, the error prompt information can include a page identifier of the target page provided by the target application program on the first page. When the target trigger action corresponds to the target trigger area in the N target trigger areas, the error prompt information can further include indication information used to indicate the element corresponding to the target trigger area, for example, can further include the position information corresponding to the target trigger area on the first page or an element identifier of the element corresponding to the target trigger area.

[0063] The process of performing one inspection on a page provided by the target application program is described in the foregoing step S201 to step S215. In a specific technical scenario, the foregoing process can be performed a plurality of times to automatically complete overall inspection on the target application program.

[0064] Referring to the foregoing description, the agent repeatedly and continuously observes the state, determines required action decision information, and receives a feedback, so that the agent can continuously update the policy of the agent. An ultimate goal is to learn a policy to maximize accumulated obtained environmental rewards. After the policy of the agent is fully updated, the agent can provide better action decision information in different states. Therefore, based on the foregoing step S201 to step S215, the following step S217 and step S219 can further be used to update the reinforcement learning model.

[0065] Step S217: Add a training sample to a preset playback buffer, where the training sample includes the page embedding of the first page, the target action, reward information, and a page embedding of the second page, where the reward information includes at least the environment reward.

[0066] In some embodiments, the reward information can further include the exploration reward for performing the target action on the first page.

[0067] Step S219: When a quantity of training samples added to the playback buffer reaches a preset threshold, train the reinforcement learning model by using the training samples stored in the playback buffer. After the reinforcement learning model is trained by using the training samples

stored in the playback buffer, the playback buffer can be re-initialized, so that a new training sample is added to the playback buffer.

[0068] In some embodiments, when the reward information further includes the exploration reward for performing the target action on the first page, an algorithm used to output the foregoing exploration reward can be further correspondingly updated by using a second page separately included in each training sample in the playback buffer, for example, the prediction network in the RND algorithm is correspondingly updated.

[0069] After the reinforcement learning model is updated a plurality of rounds, the reinforcement learning model can provide excellent action decision information based on a page embedding corresponding to a current page displayed by the terminal for the target application program at a current moment, so that the terminal displays a large quantity of pages provided by the target application program, thereby completing inspection of the target application program.

[0070] Based on a same concept as the foregoing method embodiment, an embodiment of this specification provides an application program inspection apparatus 400 based on a reinforcement learning model. As shown in FIG. 4, the inspection apparatus includes: a target detection unit 401, configured to determine, from a first page displayed by a terminal for a target application program, N trigger areas that are used to perform tap interaction; an embedding processing unit 403, configured to determine a page embedding of the first page based on the N trigger areas; a model invoking unit 405, configured to input the page embedding to a reinforcement learning model, so that the reinforcement learning model outputs action decision information corresponding to action space, where the action space includes several preset actions and N trigger actions that are respectively used to trigger the N trigger areas; an action execution unit 407, configured to perform a target action on the first page based on the action decision information, so that the terminal displays a second page; and a reward calculation unit 409, configured to obtain an environment reward for performing the target action on the first page, where a value taken by the environment reward when the second page is a newly appeared page on the terminal is greater than a value taken by the environment reward when the second page is not a newly appeared page on the terminal, and the environment reward is used to update the reinforcement learning model.

[0071] For ease of description, the above-mentioned apparatus is described by dividing the apparatus into various units/modules based on functions. Certainly, during implementation of this specification, functions of all modules/units can be implemented in one or more pieces of software and/or hardware.

[0072] 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.

[0073] 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.

[0074] 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.

[0075] 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.

[0076] 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.

[0077] 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.

[0078] 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.

[0079] 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.

[0080] 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.

[0081] 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.

[0082] 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.

[0083] Embodiments of this specification are described in a progressive manner, and for the same and similar parts of the embodiments, references can be made to each other. Each embodiment focuses on a difference from other embodiments. Particularly, the system embodiment is basically similar to the method embodiment, and therefore is briefly described. For related parts, references can be made to partial descriptions in the method embodiments.

[0084] The above-mentioned descriptions are merely embodiments of this specification, and are not intended to limit this specification. A person skilled in the art can make various modifications and variations 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. An application program inspection method based on reinforcement learning, wherein the method comprises:

determining, from a first page displayed by a terminal for a target application program, N trigger areas that are used to perform tap interaction;

determining a page embedding of the first page based on the N trigger areas;

inputting the page embedding to a reinforcement learning model, so that the reinforcement learning model outputs action decision information corresponding to action space, wherein the action space comprises several preset actions and N trigger actions that are respectively used to trigger the N trigger areas;

performing a target action on the first page based on the action decision information, so that the terminal displays a second page; and

obtaining an environment reward for performing the target action on the first page, wherein a value taken by the environment reward when the second page is a newly appeared page on the terminal is greater than a value taken by the environment reward when the second page is not a newly appeared page on the terminal, and the environment reward is used to update the reinforcement learning model.

2. The method according to claim 1, wherein the determining, from a first page displayed by a terminal for a target application program, N trigger areas that are used to perform tap interaction comprises: inputting the first page to a pre-trained deep learning model, so that the deep learning model detects, on the first page, the N trigger areas used to perform tap interaction.

3. The method according to claim 1, wherein the determining a page embedding of the first page based on the N trigger areas comprises:

encoding image areas on the first page that respectively correspond to the N trigger areas to obtain image embeddings respectively corresponding to the N trigger areas;

encoding position information that is on the first page and that respectively corresponds to the N trigger areas to obtain position embeddings respectively corresponding to the N trigger areas; and

combining the image embeddings and the position embeddings that respectively correspond to the N trigger areas to obtain the page embedding.

4. The method according to claim 1, wherein the action decision information is used to indicate an execution probability corresponding to each action in the action space, or the action decision information is used to indicate the target action expected to be performed.

5. The method according to claim 1, wherein the performing a target action on the first page based on the action decision information comprises: when the target action does not belong to the several preset actions, performing the target action based on position information that is on the first page and that is of a target trigger area corresponding to the target action.

6. The method according to claim 1, wherein when the second page is not a newly appeared page and is similar to the first page or a preset reference page, a value of the environment reward is a first preset value; and when the second page is not a newly appeared page and is not similar to the first page or the preset reference page, a value of the environment reward is a second preset value, wherein the first preset value is less than the second preset value.

7. The method according to claim 1, wherein the method further comprises: when the second page is similar to the first page or a preset reference page that appears on the terminal, generating error prompt information corresponding to the first page.

8. The method according to claim 1, wherein the method further comprises: obtaining a page code used to represent the second page, and determining, based on the page code, whether the second page is a newly appeared page on the terminal.

9. The method according to claim 8, wherein the obtaining a page code used to represent the second page comprises:

obtaining page dump information that is on the terminal and that corresponds to the second page;

extracting, from the page dump information, content information provided by the target application program;

determining, from the content information, coordinate information respectively corresponding to several target elements that carry a text; and

determining the page code of the second page based on the coordinate information respectively corresponding to the several target elements.

10. The method according to claim 1, wherein the method further comprises: obtaining an exploration reward for performing the target action on the first page, wherein the exploration reward is used to update the reinforcement learning model.

11. The method according to any one of claims 1 to 10, wherein the method further comprises:

adding a training sample to a playback buffer, wherein the training sample comprises the page embedding of the first page, the target action, reward information, and a page embedding of the second page, wherein the reward information comprises at least the environment reward; and

when a quantity of training samples added to the playback buffer reaches a preset threshold, training the reinforcement learning model by using the training samples stored in the playback buffer.

12. 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 executes the computer program to implement the following steps:

determining, from a first page displayed by a terminal for a target application program, N trigger areas that are used to perform tap interaction;

determining a page embedding of the first page based on the N trigger areas;

inputting the page embedding to a reinforcement learning model, so that the reinforcement learning model outputs action decision information corresponding to action space, wherein the action space comprises several preset actions and N trigger actions that are respectively used to trigger the N trigger areas;

performing a target action on the first page based on the action decision information, so that the terminal displays a second page; and

obtaining an environment reward for performing the target action on the first page, wherein a value taken by the environment reward when the second page is a newly appeared page on the terminal is greater than a value taken by the environment reward when the second page is not a newly appeared page on the terminal, and the environment reward is used to update the reinforcement learning model.

13. The computer system according to claim 12, wherein the determining, from a first page displayed by a terminal for a target application program, N trigger areas that are used to perform tap interaction comprises: inputting the first page to a pre-trained deep learning model, so that the deep learning model detects, on the first page, the N trigger areas used to perform tap interaction.

14. The computer system according to claim 12, wherein the determining a page embedding of the first page based on the N trigger areas comprises:

encoding image areas on the first page that respectively correspond to the N trigger areas to obtain image embeddings respectively corresponding to the N trigger areas;

encoding position information that is on the first page and that respectively corresponds to the N trigger areas to obtain position embeddings respectively corresponding to the N trigger areas; and

combining the image embeddings and the position embeddings that respectively correspond to the N trigger areas to obtain the page embedding.

15. The computer system according to claim 12, wherein the action decision information is used to indicate an execution probability corresponding to each action in the action space, or the action decision information is used to indicate the target action expected to be performed.

16. The computer system according to claim 12, wherein the performing a target action on the first page based on the action decision information comprises: when the target action does not

belong to the several preset actions, performing the target action based on position information that is on the first page and that is of a target trigger area corresponding to the target action.

17. The computer system according to claim 12, wherein when the second page is not a newly appeared page and is similar to the first page or a preset reference page, a value of the environment reward is a first preset value; and when the second page is not a newly appeared page and is not similar to the first page or the preset reference page, a value of the environment reward is a second preset value, wherein the first preset value is less than the second preset value.

18. The computer system according to claim 12, wherein the processor executes the computer program to further implement the following step: when the second page is similar to the first page or a preset reference page that appears on the terminal, generating error prompt information corresponding to the first page.

19. An application program inspection apparatus based on reinforcement learning, wherein the apparatus comprises:

a target detection unit, configured to determine, from a first page displayed by a terminal for a target application program, N trigger areas that are used to perform tap interaction;

an embedding processing unit, configured to determine a page embedding of the first page based on the N trigger areas;

a model invoking unit, configured to input the page embedding to a reinforcement learning model, so that the reinforcement learning model outputs action decision information corresponding to action space, wherein the action space comprises several preset actions and N trigger actions that are respectively used to trigger the N trigger areas;

an action execution unit, configured to perform a target action on the first page based on the action decision information, so that the terminal displays a second page; and

a reward calculation unit, configured to obtain an environment reward for performing the target action on the first page, wherein a value taken by the environment reward when the second page is a newly appeared page on the terminal is greater than a value taken by the environment reward when the second page is not a newly appeared page on the terminal, and the environment reward is used to update the reinforcement learning model.

20. A computer-readable nonvolatile storage medium, wherein the storage medium stores a computer program, and the computer program is executed by a processor to implement the method according to any one of claims 1 to 11.