

VIRTUAL TABULAR DATA GENERATION METHOD AND SERVER PERFORMING THE SAME

TECHNICAL FIELD

[0001] The disclosure relates to a virtual tabular data generation method and a server performing the same. More particularly, the disclosure relates to a method of generating highly specialized virtual data related to a topic desired by a user through prompting for a deep-learning model.

BACKGROUND

[0002] The contents set forth in this section merely provide background information on the present embodiments and do not constitute prior art.

[0003] Recently, as financial institutions or electronic financial companies offer financial products and services via computing devices, financial transactions conducted online without users having to meet face-to-face with employees of financial institutions or electronic financial companies are on the rise. As such non-face-to-face transactions increase, the importance of technology for accurately and quickly processing users' sensitive information (e.g., personal information, financial information, etc.) is increasing day by day.

[0004] In addition, various services that incorporate AI and machine learning technologies are being rapidly developed recently in the financial industry as well. In this case, in order for users or developers to output the data they want, a large amount of training data for training a deep-learning module (e.g., an LLM (large language model) or a PLM (pre-trained language model)) is needed.

[0005] However, since financial institutions handle documents related to sensitive user information, data containing users' sensitive information data cannot be used as training data as it is. To resolve this issue, financial institutions were able to use roundabout methods of replacing users' sensitive information data with data that had already been made public, but there was a problem that the public data available in this case had a low relevance with actual user data and could not contribute to improving the performance of the deep-learning module.

SUMMARY

[0006] It is an object of the present disclosure to provide a method of generating virtual tabular data that can generate highly accurate data available to improve the performance of business models for new services by replacing data containing sensitive information of customers.

[0007] In addition, it is an object of the present disclosure to provide a method of generating virtual tabular data that can verify and correct generated data so that the generated data has accuracy and diversity higher than a predetermined reference value and can reflect the characteristics of sensitive information data.

[0008] Further, it is an object of the present disclosure to provide a method of generating virtual tabular data that can generate highly accurate data that fits a data schema requested by a user by automatically updating a prompt for generating virtual tabular data over a plurality of times.

[0009] The objects of the present disclosure are not limited to the objects mentioned above, and other objects and advantages of the present disclosure that have not been mentioned can be understood by the following description and will be more clearly understood by the embodiments of the present disclosure. Moreover, it will be readily appreciated that the objects and advantages of the present disclosure can be realized by the means set forth in the claims and combinations thereof.

[0010] According to some aspects of the disclosure, a method of generating virtual tabular data, performed on a server using a deep-learning module, comprises: generating a first prompt for generating a table schema, calibrating a table schema by comparing the table schema generated based on the first prompt with a predefined reference table schema, generating a second prompt by referring to the calibrated table schema, generating table condition data for first tabular data generated based on the second prompt, generating a third prompt by referring to the table condition data and the calibrated table schema, and deriving final tabular data through a verification operation on second tabular data generated based on the third prompt.

[0011] According to some aspects, the calibrating the table schema comprises: applying the first prompt to the deep-learning module and receiving the table schema as an output of the deep-learning module; comparing the table schema with the predefined reference table schema; and calibrating the table schema to include an item included in the predefined reference table schema if the table schema does not include the item.

[0012] According to some aspects, the generating the first prompt comprises: receiving a data generation request including information on tabular data desired by a user from a user terminal linked with the server; and generating the first prompt based on information included in the data generation request.

[0013] According to some aspects, the generating the second prompt comprises generating the second prompt by referring to the calibrated table schema and the data generation request together, and the generating the third prompt comprises generating the third prompt by referring to all of the table condition data, the calibrated table schema, and the data generation request.

[0014] According to some aspects, the generating the table condition data comprises: applying the second prompt to the deep-learning module and receiving the first tabular data as an output of the deep-learning module; and generating the table condition data by comparing each column included in the first tabular data with predefined condition data.

[0015] According to some aspects, the table condition data comprises a unary constraint or a binary constraint for each column included in tabular data, and the unary constraint refers to a condition of having one of predetermined values, and the binary constraint refers to a condition of including an operational expression with another column.

[0016] According to some aspects, the deriving the final tabular data comprises: applying the third prompt to the deep-learning module and receiving the second tabular data as an output of the deep-learning module; applying the second tabular data to a data verification module and obtaining a data evaluation result as an output of the data verification module; and determining the final tabular data based on the data evaluation result.

[0017] According to some aspects, the second tabular data comprises a greater number of example data than the first tabular data, and wherein the data verification module: performs diversity verification on the example data included in the second tabular data, and performs constraint satisfaction verification on columns to which a unary constraint or a binary constraint is applied in the example data.

[0018] According to some aspects, wherein the obtaining the data evaluation result comprises: deriving a first evaluation value for diversity of each row data or each column data included in the second tabular data; deriving a second evaluation value for whether each column data included in the second tabular data satisfies a unary constraint; deriving a third evaluation value

for whether each column data included in the second tabular data satisfies a binary constraint; and determining whether reference values for the first to third evaluation values are satisfied, and generating the data evaluation result including a result thereof.

[0019] According to some aspects, the determining the final tabular data comprises: correcting data included in the second tabular data that do not satisfy the reference values to values that satisfy the reference values; and determining the second tabular data with corrections reflected as the final tabular data.

[0020] According to some aspects, correcting the third prompt based on the data evaluation result; regenerating the second tabular data by applying the corrected third prompt to the deep-learning module; applying the regenerated second tabular data to the data verification module and re-obtaining the data evaluation result as an output of the data verification module; and determining the final tabular data based on the re-obtained data evaluation result.

[0021] According to some aspects of the disclosure, a server comprises: a processor, a memory configured to load a computer program executed by the processor, and a database configured to store data generated in an execution process of the computer program, wherein the computer program comprises: generating a first prompt for generating a table schema, calibrating a table schema by comparing the table schema generated based on the first prompt with a predefined reference table schema, generating a second prompt by referring to the calibrated table schema, generating table condition data for first tabular data generated based on the second prompt, generating a third prompt by referring to the table condition data and the calibrated table schema, deriving final tabular data through a verification operation on second tabular data generated based on the third prompt, and storing the final tabular data in the database.

[0022] According to some aspects, the calibrating the table schema comprises: applying the first prompt to the deep-learning module and receiving the table schema as an output of the deep-learning module; comparing the table schema with the predefined reference table schema; and calibrating the table schema to include an item included in the predefined reference table schema if the table schema does not include the item, wherein the generating the table condition data comprises: applying the second prompt to the deep-learning module and receiving the first tabular data as an output of the deep-learning module; and generating the table condition data by comparing each column included in the first tabular data with predefined condition data, and wherein the deriving the final tabular data comprises: applying the third prompt to the deep-

learning module and receiving the second tabular data as an output of the deep-learning module; applying the second tabular data to a data verification module loaded into the memory and obtaining a data evaluation result as an output of the data verification module; and determining the final tabular data based on the data evaluation result.

[0023] According to some aspects, the obtaining the data evaluation result comprises: deriving a first evaluation value for diversity of each row data or each column data included in the second tabular data; deriving a second evaluation value for whether each column data included in the second tabular data satisfies a unary constraint; deriving a third evaluation value for whether each column data included in the second tabular data satisfies a binary constraint; and determining whether reference values for the first to third evaluation values are satisfied, and generating the data evaluation result including a result thereof.

[0024] According to some aspects, a computer-readable recording medium having recorded thereon a program capable of executing the method set forth in any one of claims 1 to 11.

[0025] The method of generating virtual tabular data of the present disclosure can generate highly accurate virtual data that can replace data containing sensitive information on customers (hereinafter, sensitive information data), and can generate virtual data to fit the form of a table schema desired by a user. Thereby, developers or workers can improve the performance of business models by utilizing highly accurate virtual data to test or train business models applicable to new services.

[0026] In addition, the virtual tabular data generated through the present disclosure can generate virtual data that can have accuracy and diversity higher than the predetermined reference value and can reflect the characteristics of sensitive information data through the data verification and correction process. Thereby, the present disclosure can solve the security issue of sensitive information data, and at the same time, generate data in a form desired by the user and utilize this to operate business models.

[0027] Furthermore, the present disclosure can generate highly accurate tabular data requested by a user by automatically updating the prompts for generating virtual tabular data over a plurality of times. Thereby, the present disclosure can increase the work efficiency of developers or workers and optimize the resources required for developing new services or analyzing existing

services by omitting or automating the data collection process required for developing new services.

[0028] In addition to the contents described above, specific effects of the present disclosure will be described together while describing specific details for carrying out the present disclosure below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0029] FIG. 1 is a conceptual diagram for describing a virtual tabular data generation system according to some embodiments of the present disclosure.

[0030] FIG. 2 is a block diagram for describing the components of a server according to some embodiments of the present disclosure.

[0031] FIG. 3 is a flowchart for describing a method of generating virtual tabular data according to some embodiments of the present disclosure.

[0032] FIG. 4 is a diagram for describing one example of a process of generating tabular data performed in each step of FIG. 3.

[0033] FIG. 5 is a flowchart for describing step S100 of FIG. 3.

[0034] FIG. 6 is a block diagram for describing the flowchart of FIG. 5.

[0035] FIG. 7 is a diagram for describing a neural network model applicable to the deep-learning module of FIG. 6.

[0036] FIG. 8 is a flowchart for describing step S200 of FIG. 3.

[0037] FIG. 9 is a block diagram for describing the flowchart of FIG. 8.

[0038] FIG. 10 is a flowchart for describing step S300 of FIG. 3.

[0039] FIG. 11 is a block diagram for describing the flowchart of FIG. 10.

[0040] FIG. 12 is a flowchart for describing one embodiment of steps S323 and S325 of FIG. 10.

[0041] FIG. 13 is a block diagram for describing the flowchart of FIG. 12.

[0042] FIG. 14 is a flowchart for describing another embodiment of step S325 of FIG. 10.

[0043] FIG. 15 is a block diagram for describing the flowchart of FIG. 14.

[0044] FIG. 16 is a diagram for describing hardware implementation of a device or system that performs a method of generating virtual tabular data according to some embodiments of the present disclosure.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0045] The terms or words used in the disclosure and the claims should not be construed as limited to their ordinary or lexical meanings. They should be construed as the meaning and concept in line with the technical idea of the disclosure based on the principle that the inventor can define the concept of terms or words in order to describe his/her own inventive concept in the best possible way. Further, since the embodiment described herein and the configurations illustrated in the drawings are merely one embodiment in which the disclosure is realized and do not represent all the technical ideas of the disclosure, it should be understood that there may be various equivalents, variations, and applicable examples that can replace them at the time of filing this application.

[0046] Although terms such as first, second, A, B, etc. used in the description and the claims may be used to describe various components, the components should not be limited by these terms. These terms are only used to differentiate one component from another. For example, a first component may be referred to as a second component, and similarly, a second component may be referred to as a first component, without departing from the scope of the disclosure. The term ‘and/or’ includes a combination of a plurality of related listed items or any item of the plurality of related listed items.

[0047] The terms used in the description and the claims are merely used to describe particular embodiments and are not intended to limit the disclosure. Singular forms are intended to include plural forms unless the context clearly indicates otherwise. In the application, terms such as “comprise,” “comprise,” “have,” etc. should be understood as not precluding the possibility of existence or addition of features, numbers, steps, operations, components, parts, or combinations thereof described herein.

[0048] Unless otherwise defined, the phrases “A, B, or C,” “at least one of A, B, or C,” or “at least one of A, B, and C” may refer to only A, only B, only C, both A and B, both A and C, both B and C, all of A, B, and C, or any combination thereof.

[0049] Unless being defined otherwise, all terms used herein, including technical or scientific terms, have the same meaning as commonly understood by those skilled in the art to which the disclosure pertains.

[0050] Terms such as those defined in commonly used dictionaries should be construed as having a meaning consistent with the meaning in the context of the relevant art, and are not to be construed in an ideal or excessively formal sense unless explicitly defined in the application. In addition, each configuration, procedure, process, method, or the like included in each embodiment of the disclosure may be shared to the extent that they are not technically contradictory to each other.

[0051] Further, machine learning is a field of developing algorithms and technologies that allow computers to be trained based on data as one field of AI, is a core technology in various fields such as data processing, image recognition, voice recognition, and Internet search, and exhibits excellent performance in deriving relevant information and generating new data.

[0052] Hereinafter, a method of generating virtual tabular data and a server performing the same according to some embodiments of the present disclosure will be discussed with reference to FIGS. 1 to 16.

[0053] FIG. 1 is a conceptual diagram for describing a virtual tabular data generation system according to some embodiments of the present disclosure.

[0054] Referring to FIG. 1, the virtual tabular data generation system according to some embodiments of the present disclosure includes a server 100 and a user terminal 200. In addition, the virtual tabular data generation system may further include a deep-learning server 300 in some embodiments. Here, the server 100 may operate in conjunction with the user terminal 200 or the deep-learning server 300 via a communication network 400, and transmit and receive data to and from the user terminal 200 or the deep-learning server 300.

[0055] First, the server 100 may generate data requested by the user terminal 200 by using a deep-learning model. At this time, the user terminal 200 may transfer a data generation request

including information on the type (e.g., financial data) and format (e.g., tabular data) of data desired by the user to the server 100.

[0056] For example, the data requested by the user terminal 200 may include personal information, financial activity information, etc. In this case, the financial activity information may include information on financial products such as deposits, savings, loans, etc., and information on customer spending patterns such as card payment history, etc. However, this is merely one example, and the present disclosure is not limited thereto.

[0057] In the following, an example in which the user terminal 200 has requested “financial data” in the form of tabular data from the server 100 will be described for the convenience of description.

[0058] The server 100 may generate a prompt reflecting the data generation request received from the user terminal 200, and apply it to a pre-trained deep-learning model to thereby generate virtual tabular data. At this time, the server 100 performs an operation of updating the prompt over a plurality of times in order to obtain more accurate tabular data.

[0059] Here, the “virtual tabular data” includes virtual data generated by the deep-learning model, unlike real data on particular users or anonymous data related to anonymous users. In some embodiments of the present disclosure, the data generated by the deep-learning model is not limited to tabular data.

[0060] However, a description will be given by taking an example in which the deep-learning model of the present disclosure generates virtual tabular data in the following for the convenience of description. In addition, the virtual tabular data and the tabular data will be used interchangeably herein to refer to the same meaning.

[0061] The virtual tabular data generated by the server 100 can have a very high similarity to real data (e.g., financial data including sensitive personal information), and may include realistic and specialized up-to-date financial activity information. The virtual tabular data generated by the server 100 may be widely used for conducting simulation tests on new financial products, for research on improving the performance of new business models, etc. A detailed description of a method of generating virtual tabular data performed by the server 100 will be given later.

[0062] Further, the user terminal 200 may refer to a terminal used by a user (e.g., an administrator or operator) and capable of operating an application in a wired or wireless communication

environment. The user terminal 200 may include various forms of electronic devices, such as, for example, a personal computer (PC), a laptop, a tablet PC, a mobile phone, a smartphone, and a wearable device (e.g., a watch-type terminal). Moreover, the user terminal 200 may not refer to one particular terminal but may be used as a sense to generally refer to various electronic devices used by the user.

[0063] The communication network 400 serves to connect the server 100 and the user terminal 200. In other words, the communication network 400 refers to a communication network that provides an access path so that the user terminal 200 can transmit and receive data after accessing the server 100. The communication network 400 may encompass, for instance, wired networks such as LANs (local area networks), WANs (wide area networks), MANs (metropolitan area networks), ISDNs (integrated services digital networks), or wireless networks such as wireless LANs, CDMA, Bluetooth, and satellite communications, but the scope of the present disclosure is not limited thereto.

[0064] Further, in some embodiments of the present disclosure, the server 100 may operate together with the deep-learning server 300 and generate the virtual tabular data described above. In this case, the deep-learning server 300 may be a device (or server) that specializes in operating a deep-learning model including a neural network trained to generate virtual tabular data. According to some embodiments, the deep-learning server 300 may utilize a transformer model based on an encoder-decoder structure. For example, the deep-learning server 300 may use an open-source artificial intelligence model such as ChatGPT.

[0065] Here, the server 100 and the deep-learning server 300 may each be configured in separate devices, or may also be configured to be included in the same hardware system. In this case, the server 100 or the deep-learning server 300 may each be implemented in at least one of a workstation, a data center, an Internet data center (IDC), a direct attached storage (DAS) system, a storage area network (SAN) system, a network attached storage (NAS) system, a redundant array of inexpensive disks or a redundant array of independent disks (RAID) system, and an electronic document management (EDMS) system, but the present embodiment is not limited thereto.

[0066] In other words, the method of generating virtual tabular data according to some embodiments of the present disclosure may be implemented only in the server 100, or may be implemented in the server 100 and the deep-learning server 300 together.

[0067] In addition, the management entities of the server 100 and the deep-learning server 300 may be the same or different from each other. For example, a particular financial institution may manage and operate the server 100 and the deep-learning server 300 together. As another example, the management entities of the server 100 and the deep-learning server 300 may be different from each other, in which case the deep-learning server 300 may refer to an external server such as a ChatGPT server, etc.

[0068] However, a description will be given by taking an example in which the execution entity of the method of generating virtual tabular data according to some embodiments of the present disclosure is the server 100, and the deep-learning model used in the present disclosure uses a module included in the server 100 in the following for the convenience of description. In the following, the respective components and modules of the server 100 will be described with reference to FIG. 2.

[0069] FIG. 2 is a block diagram for describing the components of a server according to some embodiments of the present disclosure.

[0070] Referring to FIG. 2, a server 100 according to some embodiments of the present disclosure includes an interface 110, a database 120, a processor 130, and a memory 140.

[0071] In this case, a prompt-generating module (hereinafter, PGM), a deep-learning module (hereinafter, DM), a calibration module (hereinafter, CM), a condition-generating module (hereinafter, CGM), a data verification module (hereinafter, VM), or a data correction module (hereinafter, DCM) may be loaded into the memory 140 and driven (or executed) by the processor 130. According to an embodiment, the data verification module VM may include a diversity verification model (hereinafter, a DVM) or a condition verification model CVM.

[0072] Each module may be stored and used in the form of a computer program in a database 120 or storage (not shown) included in the server 100. In addition, some embodiments of the present disclosure may be implemented with some of the modules described above omitted.

[0073] Specifically, the interface 110 may transfer the data received by the server 100 from the user terminal 200 or the deep-learning server 300 to other components in the server 100. The interface 110 may be connected to an input/output device, which is provided in the server 100 and is for receiving user input. In addition, the interface 110 may include various communication

modules and may perform data exchange between the user terminal 200 or the deep-learning server 300 via the communication network 400.

[0074] The database 120 performs the function of storing and managing data received via the interface 110. The database 120 may store a plurality of modules, in the form of a program, that is loaded into the memory 140 and is available. In addition, not only training data for training the deep-learning model but also weights of a neural network (i.e., neural network weights of a pre-trained deep-learning model) consisting of a plurality of layers may be stored and managed in the database 120. However, this is merely one example, and the present disclosure is not limited thereto.

[0075] The processor 130 may control at least one other component (e.g., hardware or software component) of the server 100 by executing software, and perform various data processing and operations. For example, the processor 130 may load information, commands, or data received from other components (e.g., the database 120) into the memory 140, perform operations using the loaded information, commands, or data, and store the resulting data in the database 120 or storage (not shown).

[0076] The memory 140 may load and store various data used by at least one component (e.g., the processor 130) of the server 100. For example, the data may include input data or output data for software and commands related thereto.

[0077] Therefore, the processor 130 may load and use modules or instructions related to various operations of the method of generating virtual tabular data according to some embodiments of the present disclosure on the memory 140.

[0078] Specifically, the processor 130 may generate a prompt for generating virtual tabular data desired by the user by using the prompt-generating module PGM loaded into the memory 140. Specifically, the prompt-generating module PGM may generate a first prompt for generating a table schema for the virtual tabular data, a second prompt including a calibrated table schema, and a third prompt including the calibrated table schema and table condition data.

[0079] For example, the first prompt may include a query for generating a table schema (e.g., a list and description of attributes of card payment history) and detailed conditions therefor. The second prompt may include a query for generating tabular data based on the calibrated table schema (e.g., please generate data on the card payment history, but be sure to comply with the

provided table schema). The third prompt may include a query for generating tabular data based on the calibrated table schema and conditions for particular columns of the table schema (e.g., please generate data on the card payment history, but be sure to comply with the provided table schema and comply with a unary constraint and a binary constraint). However, this is merely one example, and the present disclosure is not limited thereto.

[0080] The first to third prompts generated by the prompt-generating module PGM may be sequentially transferred to the deep-learning module DM. However, the present disclosure is not limited thereto, and the configuration and transfer sequence of each prompt may be modified and implemented in a variety of ways as well.

[0081] Then, the processor 130 may generate a table schema or tabular data based on the prompts generated by the prompt-generating module PGM by using the deep-learning module DM loaded into the memory 140. At this time, the deep-learning module DM may use various types of neural network structures. For example, the deep-learning module DM may be implemented in a neural network model such as a large language model (LLM) or a pre-trained language model (PLM). However, this is merely one example, and the present disclosure is not limited thereto.

[0082] In addition, the deep-learning module DM may be used by applying weights for a pre-trained and derived neural network thereto. As described above, the weights of the deep-learning module DM or the neural network included in the deep-learning module DM may be stored and used in the database 120. Moreover, the table schema or tabular data output from the deep-learning module DM may likewise be stored, used, and managed in the database 120.

[0083] Further, the processor 130 may calibrate the table schema generated based on the first prompt in the deep-learning module DM by using the calibration module CM loaded into the memory 140. At this time, the calibration module CM may calibrate the table schema output from the deep-learning module DM by using a reference table schema stored in advance in the database 120 or received from the user terminal 200. In the following, a description will be given by taking an example in which the calibration module CM calibrates the table schema by using the reference table schema stored in advance in the database 120, for the convenience of description. The particular calibration operation of the calibration module CM will be described later below.

[0084] In addition, the processor 130 may extract table condition data applicable to the tabular data generated based on the second prompt in the deep-learning module DM by using the condition-generating module CGM loaded into the memory 140. At this time, the condition-generating module CGM may generate the table condition data by comparing the condition data received from the user terminal 200 or stored in advance in the database 120 with the tabular data generated based on the second prompt. In the following, a description will be given by taking an example in which the condition-generating module CGM generates the table condition data by using the condition data stored in advance in the database 120 for the convenience of description.

[0085] The “table condition data” output from the condition-generating module CGM may include a unary constraint or a binary constraint for each column included in the tabular data.

[0086] Here, the “unary constraint” refers to a condition of having any one of predetermined values. For example, the unary constraint may be a condition that causes cells in a particular column to have an alternative value, such as 0 or 1, Yes or No, and approval or rejection. As another example, the unary constraint may be a condition of having any one value of three predetermined options.

[0087] In addition, the “binary constraint” refers to a condition of including an operational expression with another column. For example, the value of the first column may be determined by an operation (e.g., an operation using arithmetic operations or a predetermined function) between the value of the second column and the value of the second column. However, these are merely some examples of unary or binary constraints, and the present disclosure is not limited thereto.

[0088] Furthermore, the processor 130 may perform data evaluation on the tabular data generated based on the third prompt in the deep-learning module DM by using the data verification module VM loaded into the memory 140. At this time, the data verification module VM may perform diversity verification on example data (V1 to Vn in FIG. 4) included in the tabular data, or perform verification of constraint satisfaction on the column to which the unary constraint or binary constraint is applied in the example data. A detailed description of the operation of the data verification module VM will be given later with reference to FIGS. 10 to 15.

[0089] At this time, in one embodiment of the present disclosure, the processor 130 may transfer data evaluation results output from the data verification module VM to the prompt-generating

module PGM, and the prompt-generating module PGM may correct the third prompt based on the data evaluation results. Next, the processor 130 may repetitively perform an operation of applying the corrected third prompt to the deep-learning module DM and regenerating the tabular data.

[0090] Further, in another embodiment of the present disclosure, the processor 130 may perform correction on the data included in the tabular data generated based on the third prompt in the deep-learning module DM by using the data correction module DCM loaded into the memory 140. At this time, the data correction module DCM may perform an operation of correcting data that does not satisfy a predetermined reference value out of the tabular data to a value that satisfies the reference value, and this correction may be performed automatically according to a predetermined rule.

[0091] A detailed description of the process of verifying and correcting the tabular data generated based on the third prompt by the processor 130 will be given later with reference to FIGS. 12 to 15.

[0092] In the following, the operation of the method of generating virtual tabular data according to some embodiments of the present disclosure will be discussed in detail.

[0093] FIG. 3 is a flowchart for describing a method of generating virtual tabular data according to some embodiments of the present disclosure. FIG. 4 is a diagram for describing one example of a process of generating tabular data performed in each step of FIG. 3. In the following, a description will be given by taking an example in which the execution entity of the method of generating the virtual tabular data of the present disclosure is the server 100 or the processor 130 for the convenience of description.

[0094] Referring to FIGS. 3 and 4, in the method of generating virtual tabular data according to some embodiments of the present disclosure, the server 100 generates a first prompt for generating a table schema (S110). Here, the first prompt may include a query for generating a table schema (e.g., a list and description of attributes of card payment history).

[0095] Next, the server 100 calibrates the table schema by comparing the table schema generated based on the first prompt with a predefined reference table schema (S120). For example, S in <A1> of FIG. 4 represents the table schema generated with the first prompt as input, and S' in <A2> of FIG. 4 represents the calibrated table schema.

[0096] At this time, if the table schema generated based on the first prompt does not include an item of each column included in the predefined reference table schema, the server 100 may calibrate the table schema to include the corresponding item. A detailed description thereof will be given later with reference to FIGS. 5 and 6.

[0097] Next, the server 100 generates a second prompt by referring to the calibrated table schema (S210). Here, the second prompt may include a query for generating tabular data based on the calibrated table schema (e.g., please generate data on the card payment history, but be sure to comply with the provided table schema).

[0098] Next, the server 100 generates table condition data for first tabular data generated based on the second prompt (S220). For example, <A2> of FIG. 4 shows the first tabular data including the calibrated table schema S' and example data V1 therefor. Further, <A3> of FIG. 4 discloses columns to which table condition data (e.g., a unary constraint Cu1 or binary constraints Cb1 to Cb3) applicable to the calibrated table schema S' is applied.

[0099] Next, the server 100 generates a third prompt by referring to the table condition data and the calibrated table schema (S310). Here, the third prompt may include a query for generating tabular data based on the calibrated table schema and conditions (i.e., table condition data) for particular columns of the table schema (e.g., please generate data on the card payment history, but be sure to comply with the provided table schema and comply with a unary constraint and a binary constraint).

[0100] Next, the server 100 derives final tabular data through a verification operation on second tabular data generated based on the third prompt (S320). For example, <A3> of FIG. 4 shows the second tabular data including the calibrated table schema S' and a plurality of example data V1 to Vn to which the table condition data (e.g., the unary constraint Cu1 or binary constraints Cb1 to Cb3) is applied.

[0101] In the present disclosure, the server 100 can generate highly accurate tabular data requested by the user by automatically updating the prompts for generating virtual tabular data over a plurality of times in the process of sequentially performing each of the steps described above.

[0102] Thereby, the present disclosure can increase the work efficiency of developers or workers and optimize the resources required for developing new services or analyzing existing services by omitting or automating the data collection process required for developing new services.

[0103] In the following, a method of calibrating a table schema (i.e., step S100 of FIG. 3) according to some embodiments of the present disclosure will be discussed in detail.

[0104] FIG. 5 is a flowchart for describing step S100 of FIG. 3. FIG. 6 is a block diagram for describing the flowchart of FIG. 5. FIG. 7 is a diagram for describing a neural network model applicable to the deep-learning module of FIG. 6.

[0105] Referring to FIGS. 5 and 6, the server 100 may receive a data generation request including information on tabular data desired by the user from the user terminal 200 (S111). Here, the user terminal 200 may receive as input a data generation request including information on the type (e.g., financial data), format (e.g., tabular data), or conditions (e.g., including amounts and point columns) of data desired by the user from the user, and transfer the received data generation request to the server 100.

[0106] Next, the processor 130 of the server 100 inputs the received data generation request into the prompt-generating module PGM, and receives a first prompt generated based on the information included in the data generation request as an output of the prompt-generating module PGM (S113).

[0107] Next, the processor 130 applies the generated first prompt to the deep-learning module DM, and receives a table schema as an output of the deep-learning module DM (S121).

[0108] At this time, the deep-learning module DM used in some embodiments of the present disclosure may output a table schema or tabular data corresponding to the input prompt by using an artificial neural network pre-trained based on big data. Here, the tabular data refers to data including a table schema and example data generated according to the format of the table schema.

[0109] Specifically, the deep-learning module DM may be implemented in a neural network structure, and train the artificial neural network by using mapping values to separate parameters derived based on the input data. At this time, the deep-learning module DM may perform machine learning on the parameters input as learning factors.

[0110] In a more detailed description, a deep-learning technique, which is a kind of machine learning, goes down to a deep level and is subjected to learning in multiple stages based on data.

[0111] Deep learning refers to a set of machine learning algorithms that extract core data from a plurality of data while moving up the stages.

[0112] The deep-learning module DM may use a variety of known artificial neural network structures. For example, the deep-learning module DM may use structures such as a convolutional neural network (CNN), a recurrent neural network (RNN), a deep belief network (DBN), a graph neural network (GNN), and a transformer.

[0113] Further, the training of the artificial neural network by the deep-learning module DM may be achieved by calibrating the weights of connection lines between nodes (and calibrating bias values as well if necessary) so that a desired output is obtained for a given input. In addition, the artificial neural network may continuously update the weight values by training. Moreover, methods such as backpropagation may be used for training the artificial neural network.

[0114] In this case, the memory 140 of the server 100 may be equipped with an artificial neural network pre-trained with machine learning. In other words, the memory 140 may store data used for machine learning, result data, and the like.

[0115] In some embodiments of the present disclosure, the deep-learning module DM may include a deep-learning model based on a transformer including an encoder EN and a decoder DE or a large language model. However, these are merely some examples of the present disclosure, and the present disclosure is not limited thereto.

[0116] Referring to FIG. 7, the deep-learning module DM includes an input layer Input with prompts in the form of text as input nodes, an output layer Output with virtual tabular data as output nodes, and M hidden layers arranged between the input layer and the output layer.

[0117] Here, weights may be set for the edges that connect the nodes of the respective layers. The presence or absence of these weights or edges may be added, removed, or updated during the training process. Therefore, the weights of the nodes and edges arranged between the k input nodes and the i output nodes may be updated through the training process.

[0118] Before the deep-learning module DM performs training, all nodes and edges may be set to initial values. However, if information is input cumulatively, the weights of the nodes and

edges may be changed, and in this process, matching may be made between the parameters input as the training factors (i.e., character recognition data) and the values assigned to the output nodes (i.e., summary data).

[0119] Additionally, if a cloud server (not shown) (e.g., the deep-learning server 300) is used, the deep-learning module DM may receive and process a large number of parameters. In this case, the operation of the deep-learning module DM may be implemented in conjunction with the server 100 and/or a separate cloud server (not shown) in some embodiments of the present disclosure. Therefore, the deep-learning module DM may perform training based on an immense amount of data.

[0120] The weights of the nodes and edges between the input and output nodes constituting the deep-learning module DM may be updated by the training process of the deep-learning module DM. Moreover, the parameters output from the deep-learning module DM may be further expanded to various data, in addition to the summary data, as a matter of course.

[0121] Both semi-supervised learning and supervised learning may be used as the machine learning method used in the deep-learning module DM. In addition, the deep-learning module DM may be controlled to automatically update the artificial neural network structure for outputting more accurate tabular data after training according to the settings.

[0122] Referring again to FIGS. 5 and 6, the processor 130 compares the table schema output from the deep-learning module DM with the reference table schema stored in advance in the database 120 by using the calibration module CM (S123). Here, the reference table schema may define items that must be included in the tabular data. The reference table schema may be defined in advance by an expert, stored in the database 120, and used, or may be received separately from the user terminal 200 and used, as a matter of course.

[0123] At this time, if the table schema received from the deep-learning module DM does not include an item included in the reference table schema, the calibration module CM may calibrate the table schema to include the corresponding item.

[0124] For example, if the table schema S generated by the deep-learning module DM consists only of the items of Date, Amount, and Merchant as disclosed in <A1> of FIG. 4, and the reference table schema predefined in the database 120 consists of Date, Amount, Merchant, Inquiry Type, Online Pay, Regular Pay, Ratio of Point, and Point, then the calibration module

CM can calibrate the calibrated table schema S' to include all of the respective items of the reference table schema. However, this is merely one example and the present disclosure is not limited thereto.

[0125] In the following, a method of generating table condition data (i.e., step S200 of FIG. 3) according to some embodiments of the present disclosure will be discussed in detail.

[0126] FIG. 8 is a flowchart for describing step S200 of FIG. 3. FIG. 9 is a block diagram for describing the flowchart of FIG. 8.

[0127] Referring to FIGS. 8 and 9, the processor 130 generates a second prompt using the prompt-generating module PGM by referring to the table schema calibrated in step S100 (S211). At this time, although not explicitly shown in the drawing, the prompt-generating module PGM may generate the second prompt by referring to the calibrated table schema and the data generation request received from the user terminal 200 together. However, this is merely one example, and the present disclosure is not limited thereto.

[0128] Next, the processor 130 applies the second prompt to the deep-learning module DM and receives first tabular data as an output of the deep-learning module DM (S221). Here, the first tabular data may include first example data V1 generated according to the format of the calibrated table schema S' (<A2> of FIG. 4).

[0129] Next, the processor 130 compares each column included in the first tabular data with predefined condition data by using the condition-generating module CGM and generates table condition data (S223). At this time, the table condition data includes a unary constraint and/or a binary constraint for each column included in the tabular data.

[0130] At this time, the condition-generating module CGM may use condition data stored in advance in the database 120 or condition data separately received from the user terminal 200. Here, the condition data may be data in which a plurality of items (or columns) and whether a condition (i.e., the unary constraint or binary constraint) is applied to the corresponding items are stored in a table format. The condition-generating module CGM may generate table condition data by comparing the condition data with each column included in the first tabular data and determining which condition to apply to which column of the first tabular data.

[0131] For example, the table condition data may include a unary constraint that causes the column of Inquiry Type to have a value of either online or offline. In addition, the table condition

data may include a binary constraint that causes the columns of Amount, Rate of Point, and Point to be cross-referenced by an operational expression (see <A3> in FIG. 4). However, this is merely one example, and the present disclosure is not limited thereto.

[0132] In the following, a method of generating second tabular data and determining final tabular data through a verification operation (i.e., step S300 in FIG. 3) according to some embodiments of the present disclosure will be discussed in detail.

[0133] FIG. 10 is a flowchart for describing step S300 of FIG. 3. FIG. 11 is a block diagram for describing the flowchart of FIG. 10.

[0134] Referring to FIGS. 10 and 11, the processor 130 generates a third prompt using the prompt-generating module PGM by referring to the table schema calibrated in step S100 and the table condition data derived in step S200 (S311). At this time, although not explicitly shown in the drawing, the prompt-generating module PGM may generate the third prompt by referring to the calibrated table schema, the table condition data, and the data generation request received from the user terminal 200 together. However, this is merely one example, and the present disclosure is not limited thereto.

[0135] Next, the processor 130 applies the third prompt to the deep-learning module DM and receives second tabular data as an output of the deep-learning module DM (S321). Here, the second tabular data may include a plurality of example data V1 to Vn generated according to the format of the calibrated table schema S' (<A3> of FIG. 4). In this case, the second tabular data (<A3> of FIG. 4) may include a greater number of example data than the first tabular data (<A2> of FIG. 4).

[0136] Next, the processor 130 applies the second tabular data to the data verification module VM and obtains data evaluation results as an output of the data verification module VM (S323). At this time, the data verification module VM may perform diversity verification on example data included in the second tabular data, and perform verification of constraint satisfaction on the column to which the unary constraint or binary constraint is applied in the example data. Therefore, the data evaluation results may include a first evaluation value for diversity, a second evaluation value for whether the unary constraint is satisfied, and/or a third evaluation value for whether the binary constraint is satisfied.

[0137] Next, the processor 130 determines final tabular data based on the obtained data evaluation results (S325).

[0138] In the following, a process of determining the final tabular data based on the data evaluation results in some embodiments of the present disclosure will be described in detail.

[0139] FIG. 12 is a flowchart for describing one embodiment of steps S323 and S325 of FIG. 10. FIG. 13 is a block diagram for describing the flowchart of FIG. 12.

[0140] Referring to FIGS. 12 and 13, the processor 130 first transfers the second tabular data obtained from the deep-learning module DM to the data verification module VM.

[0141] At this time, the data verification module VM derives a first evaluation value for the diversity of each row data or each column data included in the second tabular data by using the diversity verification model DVM (S323a).

[0142] Specifically, in one embodiment, the data verification module VM may derive it by concatenating all the values of each row data included in the second tabular data (e.g., using a concat function), embedding the concatenated values in a particular domain, then extracting the uniformity of the embedded values, and normalizing them using a normalization function (e.g., a Sigmoid function). Thereby, the data verification module VM may derive the diversity between instances included in each row data as the first evaluation value.

[0143] Further, in another embodiment, the data verification module VM may derive the diversity of each column data included in the second tabular data (i.e., the diversity between attributes in an instance) as the first evaluation value. Here, the first evaluation value can be derived by the following <Mathematical Expression 1>:

<Mathematical Expression 1>

$$\mathcal{H} = - \sum_{i \in I} p_i * \log p_i$$

[0144] Here, $\mathcal{H}$ denotes the first evaluation value representing the degree of diversity of the data, I denotes a set of unique values corresponding to a particular column, i denotes a value included in the corresponding set, and p_i denotes the probability that each i appears.

[0145] In addition, the data verification module VM derives a second evaluation value for whether each column data included in the second tabular data satisfies the unary constraint by using the condition verification model CVM (S323b). Here, the second evaluation value can be derived by the following <Mathematical Expression 2>:

<Mathematical Expression 2>

$$\rho = \frac{1}{K^u} \sum_{c_j^u \in C^u} \mathbb{1}(\Psi(\hat{v}_{\bullet, c_j^u}))$$

[0146] Here, ρ denotes the second evaluation value for whether the unary constraint is satisfied, K^u denotes the number of unary constraint items, C^u denotes a set of values to which the unary constraint is applied, Ψ denotes a unary operator that checks the existence of a predetermined unary item, and c_j^u denotes each value to which the unary constraint is applied.

[0147] In addition, the data verification module VM derives a third evaluation value for whether each column data included in the second tabular data satisfies the binary constraint by using the condition verification model CVM (S323c). Here, the third evaluation value can be derived by the following <Mathematical Expression 3>:

<Mathematical Expression 3>

$$\tau = \frac{1}{K^b} \sum_{c_j^b \in C^b} \mathbb{1}(c_j^b)$$

[0148] Here, τ denotes the third evaluation value for whether the binary constraint is satisfied, K^b denotes the number of binary constraint items, C^b denotes a set of values to which the binary constraint is applied, and c_j^b denotes each value to which the binary constraint is applied.

[0149] Next, the data verification module VM determines whether the reference values for the first to third evaluation values are satisfied, and generates data evaluation results including the results thereof (S323d). At this time, the data verification module VM determines whether the

first to third evaluation values and each evaluation value satisfy predetermined reference values (or thresholds), and generates data evaluation results including the results thereof.

[0150] At this time, the steps S323a to S323c described above may be performed in sequence or in parallel in the data verification module VM, and may be implemented by omitting or modifying some of them. In addition, the data verification module VM may of course perform step S323d based on the evaluation values derived in the steps described above.

[0151] Next, the processor 130 checks whether the first to third evaluation values satisfy predetermined reference values by referring to the data evaluation results (S323e). At this time, the processor 130 may determine whether the reference values are satisfied based on whether each of the first to third evaluation values satisfies all of the respective predetermined reference values, or whether a predetermined number or more of evaluation values of the first to third evaluation values satisfy the respective reference values.

[0152] If the first to third evaluation values do not satisfy the reference values, the processor 130 may perform an operation of regenerating the second tabular data. At this time, the processor 130 may regenerate all or part of the second tabular data.

[0153] If only part of the second tabular data is regenerated, the processor 130 may store only the data that passed the reference values in the database 120, discard the rest of the data, and then generate the second tabular data again.

[0154] Specifically, the processor 130 transfers the data evaluation results received from the data verification module VM to the prompt-generating module PGM, and the prompt-generating module PGM corrects the third prompt based on the data evaluation results (S324a). At this time, the prompt-generating module PGM may correct the third prompt to include a conditional statement for replacement or exclusion of data that does not satisfy the reference values. However, this is merely one example, and the present disclosure is not limited thereto.

[0155] Next, the processor 130 regenerates the second tabular data by applying the corrected third prompt to the deep-learning module DM again (S324b).

[0156] Next, the processor 130 applies the regenerated second tabular data to the data verification module VM again, and re-obtains data evaluation results as an output of the data verification module VM. At this time, the operation of the data verification module VM may be substantially the same as the steps S323a to S323d described above.

[0157] Next, the processor 130 checks whether the first to third evaluation values satisfy the predetermined reference values by referring to the re-obtained data evaluation results.

[0158] On the other hand, if the first to third evaluation values satisfy the predetermined reference values, the processor 130 can determine the second tabular data as the final tabular data (S325a) and store the determined final tabular data in the database 120 (S325b).

[0159] In addition, the operation of generating the second tabular data described above may be performed repetitively until the final tabular data is determined, as a matter of course.

[0160] FIG. 14 is a flowchart for describing another embodiment of step S325 of FIG. 10. FIG. 15 is a block diagram for describing the flowchart of FIG. 14. In the following, the contents overlapping with what has been described above will be omitted, and a description will be provided mainly focusing on the differences.

[0161] Referring to FIGS. 14 and 15, the processor 130 performs steps S323a to S323d by using the data verification module VM in another embodiment of the present disclosure.

[0162] Next, the processor 130 transfers the data evaluation results received from the data verification module VM to the data correction module DCM.

[0163] At this time, the data correction module DCM performs correction on data that does not satisfy the reference value included in the second tabular data (S424). Specifically, the data correction module DCM corrects values that do not satisfy the reference value to values that satisfy the reference value. At this time, the data correction module DCM can correct values that do not satisfy the reference value to values that satisfy the reference value according to predetermined rules.

[0164] Next, the processor 130 may determine the second tabular data reflecting the corrections performed by the data correction module DCM as the final tabular data (S425a), and store the determined final tabular data in the database 120 (S425b).

[0165] Therefore, the virtual tabular data generated through the method described above can generate virtual data that can have accuracy and diversity higher than the predetermined reference value and can reflect the characteristics of sensitive information data of customers through the data verification and correction process. Thereby, the present disclosure can solve the security

issue of sensitive information data, and at the same time, generate data in a form desired by the user and utilize this to operate business models.

[0166] In summary, some embodiments of the present disclosure can automatically generate highly accurate virtual data that can replace data containing sensitive information on customers, and can generate virtual data to fit the form of a table schema desired by the user. Thereby, developers or workers can use the present disclosure to test or train business models applicable to new services, and can instantly generate and utilize highly accurate virtual data.

[0167] FIG. 16 is a diagram for describing hardware implementation of a device or system that performs a method of generating virtual tabular data according to some embodiments of the present disclosure.

[0168] Referring to FIG. 16, a server 100 that performs the method of generating virtual tabular data according to some embodiments of the present disclosure may be implemented in an electronic device 1000. The electronic device 1000 may include a processor 1010, an input/output device (I/O) 1020, a memory 1030, an interface 1040, a storage 1050, and a bus 1060. The processor 1010, the input/output device 1020, the memory 1030, the interface 1040, and/or the storage 1050 may be coupled to each other via the bus 1060. The bus 1060 corresponds to a path through which data is moved.

[0169] Specifically, the processor 1010 may include at least one of a CPU (central processing unit), an MPU (microprocessor unit), an MCU (microcontroller unit), a GPU (graphic processing unit), a microprocessor, a digital signal processor, a microcontroller, an application processor (AP), and logic devices capable of performing functions similar thereto.

[0170] The input/output device 1020 may include at least one of a keypad, a keyboard, a touchscreen, and a display device.

[0171] The memory 1030 may be loaded with data and/or programs, etc. In this case, the memory 1030 is an operating memory for improving the operation of the processor 1010 and may include high-speed DRAM and/or SRAM, etc. The memory 1030 may include one or more volatile memory devices such as DDR SDRAM (double data rate static DRAM) and SDR SDRAM (single data rate SDRAM), and/or one or more nonvolatile memory devices such as EEPROM (electrically erasable programmable ROM), and flash memory.

[0172] The interface 1040 may perform the function of transmitting data to or receiving data from a communication network. The interface 1040 may be of a wired or wireless form. For example, the interface 1040 may include an antenna or a wired/wireless transceiver.

[0173] The storage 1050 may store and preserve data and/or programs, etc. The storage 1050 may include one or more nonvolatile memory devices such as a solid-state drive (SSD), a hard drive, and flash memory. The storage 1050 in the present disclosure may store a computer program consisting of instructions for performing the method of generating virtual tabular data.

[0174] Alternatively, the server 100 and the deep-learning server 300 in accordance with embodiments of the present disclosure may each be a system formed by connecting a plurality of electronic devices 1000 to each other via a network. In such a case, each module or combinations of modules may be implemented in the electronic device 1000. However, the present embodiment is not limited thereto.

[0175] Additionally, the server 100 may be implemented in at least one of a workstation, a data center, an Internet data center (IDC), a direct-attached storage (DAS) system, a storage area network (SAN) system, a network-attached storage (NAS) system, and a RAID (redundant array of inexpensive disks, or redundant array of independent disks) system, but the present embodiment is not limited thereto.

[0176] Further, the server 100 may transmit data over a network. The network may include a network based on wired Internet technology, wireless Internet technology, and short-range communication technology. The wired Internet technology may include, for example, at least one of a local area network (LAN) and a wide area network (WAN).

[0177] The wireless Internet technology may include at least one of, for example, wireless LAN (WLAN), DMNA (Digital Living Network Alliance), WiBro (Wireless Broadband), WiMAX (World Interoperability for Microwave Access), HSDPA (High Speed Downlink Packet Access), HSUPA (High Speed Uplink Packet Access), IEEE 802.16, LTE (Long Term Evolution), LTE-A (Long Term Evolution-Advanced), WMBS (Wireless Mobile Broadband Service), and 5G NR (New Radio) technology. However, the present embodiment is not limited thereto.

[0178] The short-range communication technology may include at least one of, for example, Bluetooth, RFID (Radio Frequency Identification), IrDA (Infrared Data Association), UWB

(Ultra-Wideband), ZigBee, NFC (Near Field Communication), USC (Ultra Sound Communication), VLC (Visible Light Communication), Wi-Fi, Wi-Fi Direct, and 5G NR (New Radio). However, the present embodiment is not limited thereto.

[0179] The server 100 communicating over the network may comply with technical standards and standard communication methods for mobile communications. For example, the standard communication methods may include at least one of GSM (Global System for Mobile communication), CDMA (Code Division Multiple Access), CDMA 2000 (Code Division Multiple Access 2000), EV-DO (Enhanced Voice-Data Optimized or Enhanced Voice-Data Only), WCDMA (Wideband CDMA), HSDPA (High Speed Downlink Packet Access), HSUPA (High Speed Uplink Packet Access), LTE (Long Term Evolution), LTEA (Long Term Evolution-Advanced), and 5G NR (New Radio). However, the present embodiment is not limited thereto.

[0180] While the inventive concept has been particularly shown and described with reference to exemplary embodiments thereof, it will be understood by those of ordinary skill in the art that various changes in form and details may be made therein without departing from the spirit and scope of the inventive concept as defined by the following claims. It is therefore desired that the embodiments be considered in all respects as illustrative and not restrictive, reference being made to the appended claims rather than the foregoing description to indicate the scope of the disclosure.

CLAIMS

What is Claimed is:

1. A method of generating virtual tabular data, performed on a server using a deep-learning module, comprising:
generating a first prompt for generating a table schema;
calibrating the table schema by comparing the table schema generated based on the first prompt with a predefined reference table schema;
generating a second prompt by referring to the calibrated table schema;
generating table condition data for first tabular data generated based on the second prompt;
generating a third prompt by referring to the table condition data and the calibrated table schema; and
deriving final tabular data through a verification operation on second tabular data generated based on the third prompt.

2. The method of claim 1, wherein the calibrating the table schema comprises:
applying the first prompt to the deep-learning module and receiving the table schema as an output of the deep-learning module;
comparing the table schema with the predefined reference table schema; and
calibrating the table schema to include an item included in the predefined reference table schema if the table schema does not include the item.

3. The method of claim 1, wherein the generating the first prompt comprises:
receiving a data generation request including information on tabular data desired by a user from a user terminal linked with the server; and
generating the first prompt based on information included in the data generation request.

4. The method of claim 3, wherein the generating the second prompt comprises generating the second prompt by referring to the calibrated table schema and the data generation request together, and
the generating the third prompt comprises generating the third prompt by referring to all of the table condition data, the calibrated table schema, and the data generation request.

5. The method of claim 1, wherein the generating the table condition data comprises:
applying the second prompt to the deep-learning module and receiving the first tabular data as an output of the deep-learning module; and
generating the table condition data by comparing each column included in the first tabular data with predefined condition data.

6. The method of claim 5, wherein the table condition data comprises a unary constraint or a binary constraint for each column included in tabular data, and
the unary constraint refers to a condition of having one of predetermined values, and
the binary constraint refers to a condition of including an operational expression with another column.

7. The method of claim 1, wherein the deriving the final tabular data comprises:
applying the third prompt to the deep-learning module and receiving the second tabular data as an output of the deep-learning module;
applying the second tabular data to a data verification module and obtaining a data evaluation result as an output of the data verification module; and
determining the final tabular data based on the data evaluation result.

8. The method of claim 7, wherein the second tabular data comprises a greater number of example data than the first tabular data, and
wherein the data verification module:
performs diversity verification on the example data included in the second tabular data, and
performs constraint satisfaction verification on columns to which a unary constraint or a binary constraint is applied in the example data.

9. The method of claim 7, wherein the obtaining the data evaluation result comprises:
deriving a first evaluation value for diversity of each row data or each column data included in the second tabular data;
deriving a second evaluation value for whether each column data included in the second

tabular data satisfies a unary constraint;

deriving a third evaluation value for whether each column data included in the second tabular data satisfies a binary constraint; and

determining whether reference values for the first to third evaluation values are satisfied, and generating the data evaluation result including a result thereof.

10. The method of claim 9, wherein the determining the final tabular data comprises: correcting data included in the second tabular data that do not satisfy the reference values to values that satisfy the reference values; and

determining the second tabular data with corrections reflected as the final tabular data.

11. The method of claim 7, further comprising: correcting the third prompt based on the data evaluation result; regenerating the second tabular data by applying the corrected third prompt to the deep-learning module;

applying the regenerated second tabular data to the data verification module and re-obtaining the data evaluation result as an output of the data verification module; and

determining the final tabular data based on the re-obtained data evaluation result.

12. A server comprising: a processor; a memory configured to load a computer program executed by the processor; and a database configured to store data generated in an execution process of the computer program,

wherein the computer program comprises: generating a first prompt for generating a table schema; calibrating a table schema by comparing the table schema generated based on the first prompt with a predefined reference table schema; generating a second prompt by referring to the calibrated table schema; generating table condition data for first tabular data generated based on the second prompt; generating a third prompt by referring to the table condition data and the calibrated table schema;

deriving final tabular data through a verification operation on second tabular data generated based on the third prompt; and
storing the final tabular data in the database.

13. The server of claim 12, wherein the calibrating the table schema comprises:
applying the first prompt to the deep-learning module and receiving the table schema as an output of the deep-learning module;
comparing the table schema with the predefined reference table schema; and
calibrating the table schema to include an item included in the predefined reference table schema if the table schema does not include the item,
wherein the generating the table condition data comprises:
applying the second prompt to the deep-learning module and receiving the first tabular data as an output of the deep-learning module; and
generating the table condition data by comparing each column included in the first tabular data with predefined condition data, and
wherein the deriving the final tabular data comprises:
applying the third prompt to the deep-learning module and receiving the second tabular data as an output of the deep-learning module;
applying the second tabular data to a data verification module loaded into the memory and obtaining a data evaluation result as an output of the data verification module; and
determining the final tabular data based on the data evaluation result.

14. The server of claim 13, wherein the obtaining the data evaluation result comprises:
deriving a first evaluation value for diversity of each row data or each column data included in the second tabular data;
deriving a second evaluation value for whether each column data included in the second tabular data satisfies a unary constraint;
deriving a third evaluation value for whether each column data included in the second tabular data satisfies a binary constraint; and
determining whether reference values for the first to third evaluation values are satisfied, and generating the data evaluation result including a result thereof.

15. A computer-readable recording medium having recorded thereon a program capable of executing the method set forth in any one of claims 1 to 11.