TEXT DATA PROCESSING METHOD, TEXT REPRESENTATION METHOD, TEXT CLUSTERING METHOD, AND DEVICE

[0001] The present disclosure claims priority to Chinese Patent Application No. 202310955488.2, entitled "TEXT DATA PROCESSING METHOD, TEXT REPRESENTATION METHOD, TEXT CLUSTERING METHOD, AND DEVICE" and filed with the China National Intellectual Property Administration on July 31, 2023, which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

[0002] The present disclosure relates to computer technologies, and in particular, to a text data processing method, a text representation method, a text clustering method, and a device.

BACKGROUND

[0003] Text representation, as a basic capability in the field of natural language processing, is widely applied to downstream tasks such as text classification, clustering, retrieval, question answering, and recommendation. High-quality text representation is crucial for effects of the downstream tasks.

[0004] Conventional text representation methods, such as one-hot encoding and a bag-of-words model, have problems such as the curse of dimensionality and information loss, making it difficult to accurately extract semantic information. With the development of deep learning technologies, text representation methods based on deep learning models have become a hot topic, effectively resolving the problems of the conventional text representation methods and achieving good effects in various tasks. In recent years, the emergence of large-scale pre-trained language models (such as the Bidirectional Encoder Representations from Transformers (BERT) series of models and other Large Language Models (LLMs)) has provided a stronger model foundation. A text representation model trained based on the pre-trained language model is far superior to a conventional text representation method. However, as a general model, the pre-trained language model has a training target greatly different from a target of the text representation model during a pre-training stage.

[0005] To improve text representation quality, large-scale high-quality manually annotated data sets are usually used, to perform strong supervision training on the pre-trained language model to improve a text representation capability of the model. However, manual data annotation is time-consuming and labor-intensive, and it is difficult to obtain large-scale high-quality annotated data to train the model. As a result, an obtained text representation model has a poor text representation capability and low text representation quality, which further leads to a poor effect of a downstream processing task.

SUMMARY

[0006] The present disclosure provides a text data processing method, a text representation method, a text clustering method, and a device, to resolve a problem of a poor effect of a downstream processing task caused by a poor text representation capability and low text representation quality of a text representation model.

[0007] According to a first aspect, the present disclosure provides a text data processing method, including:

performing level classification on text data used for training, to obtain a plurality of training sets of different data levels; and

performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels, to obtain a text representation model, where the text representation model is used for generating a text representation of input text.

[0008] According to a second aspect, the present disclosure provides a text representation method, applied to a server and including:

receiving an invoking request from an end-side device for a text representation model, where the invoking request includes to-be-processed text information;

inputting the text information into the text representation model for encoding, to obtain a text representation of the text information, where the text representation model is obtained through performing level classification on text data used for training to obtain a plurality of training sets of different data levels, and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels; and

outputting the text representation of the text information to the end-side device.

[0009] According to a third aspect, the present disclosure provides a text clustering method,

including:

obtaining a plurality of to-be-clustered text segments;

separately inputting the plurality of text segments into a text representation model for encoding, to generate text representations of the text segments, where the text representation model is obtained through performing level classification on text data used for training to obtain a plurality of training sets of different data levels, and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels; and

performing clustering processing according to the text representations of the text segments, to obtain a text clustering result.

[0010] According to a fourth aspect, the present disclosure provides a text data processing apparatus, including:

a classifying module, configured to perform level classification on text data used for training, to obtain a plurality of training sets of different data levels; and

a training module, configured to perform multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels, to obtain a text representation model, where the text representation model is used for generating a text representation of input text.

[0011] According to a fifth aspect, the present disclosure provides a text representation apparatus, used in a server and including:

a receiving module, configured to receive an invoking request from an end-side device for a text representation model, where the invoking request includes to-be-processed text information;

an encoding module, configured to input the text information into the text representation model for encoding, to obtain a text representation of the text information, where the text representation model is obtained through performing level classification on text data used for training to obtain a plurality of training sets of different data levels, and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels; and

an outputting module, configured to output the text representation of the text information to the end-side device.

[0012] According to a sixth aspect, the present disclosure provides a text clustering apparatus, including:

an obtaining module, configured to obtain a plurality of to-be-clustered text segments;

an encoding module, configured to separately input the plurality of text segments into a text representation model for encoding, to generate text representations of the text segments, where the text representation model is obtained through performing level classification on text data used for training to obtain a plurality of training sets of different data levels, and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels; and

a clustering module, configured to perform clustering processing according to the text representations of the text segments, to obtain a text clustering result.

[0013] According to a seventh aspect, the present disclosure provides an electronic device, including: a processor and a memory communicatively connected to the processor, where the memory stores computer execution instructions; and the processor executes the computer execution instructions stored in the memory, to implement the method according to any one of the foregoing aspects.

[0014] According to an eighth aspect, the present disclosure provides a computer-readable storage medium, storing computer execution instructions, where when executed by a processor, the computer execution instructions are used for implementing the method according to any one of the foregoing aspects.

[0015] According to a ninth aspect, the present disclosure provides a computer program product, including a computer program, where when the computer program is executed by a computer, the method according to any one of the foregoing aspects is implemented.

[0016] According to the text data processing method, the text representation method, the text clustering method, and the device provided in the present disclosure, level classification is performed on the text data used for training, to obtain the plurality of training sets of different data levels; and multi-stage training is performed on the pre-trained model by using the configured multi-stage training policy and the plurality of training sets of different data levels, to obtain the text representation model. Compared with performing one-time (one-stage) training on the pre-trained model based on all training data, performing multi-stage training on the pre-trained model based on the plurality of training sets of different data levels can better improve a representation capability of the pre-trained model, thereby improving a text representation capability of the trained text representation model, improving text representation quality, and further improving an effect of a downstream text processing task.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] Accompanying drawings herein, which are incorporated into the specification and constitute a part of this specification, show embodiments that conform to the present disclosure, and are used for describing a principle of the present disclosure together with this specification.

FIG. 1 is a schematic diagram of an exemplary system architecture to which the present disclosure is applicable;

FIG. 2 is a flowchart of a text data processing method according to an exemplary embodiment of the present disclosure;

FIG. 3 is a framework diagram of a text data processing method according to an exemplary embodiment of the present disclosure;

FIG. 4 is a flowchart of a text data processing method according to another exemplary embodiment of the present disclosure;

FIG. 5 is a flowchart of constructing a weakly supervised training set according to an exemplary embodiment of the present disclosure;

FIG. 6 is a flowchart of a text representation method according to an exemplary embodiment of the present disclosure;

FIG. 7 is a flowchart of a text clustering method according to an exemplary embodiment of the present disclosure;

FIG. 8 is a flowchart of a text retrieval method according to an exemplary embodiment of the present disclosure; and

FIG. 9 is a schematic structural diagram of an electronic device according to an embodiment of the present disclosure.

[0018] The foregoing accompanying drawings illustrate specific embodiments of the present disclosure, and more detailed description is provided below. The accompanying drawings and text descriptions are not intended to limit the scope of the idea of the present disclosure in any manner, but are intended to describe the concept of the present disclosure to a person skilled in the art with reference to particular embodiments.

DESCRIPTION OF EMBODIMENTS

[0019] Exemplary embodiments are described in detail herein, and examples thereof are shown in the accompanying drawings. When the following descriptions are made with reference to the accompanying drawings, unless otherwise indicated, the same numbers in different accompanying drawings represent the same or similar elements. Implementations described in the following exemplary embodiments do not represent all implementations that

are consistent with the present disclosure. On the contrary, the implementations are merely examples of apparatuses and methods that are described in detail in the appended claims and that are consistent with some aspects of the present disclosure.

[0020] First, terms involved in the present disclosure are explained.

[0021] Text representation (Text Embedding): It is text vector representation, which refers to a representation method in which a text is converted into a vector form. In this representation method, mathematical encoding is performed on the text, to convert the text into a set of consecutive numerical values, to form a vector representation.

[0022] Transformer: It is an attention mechanism-based neural network model, configured to process a natural language processing task. It includes a plurality of layers of encoders and decoders. The encoder converts an input sequence into a series of hidden states, and focuses on different parts of the input sequence through an attention mechanism. The decoder uses a similar mechanism, but further uses a self-attention mechanism to ensure smoothness and consistency of an output sequence.

[0023] Contrastive learning: It is a machine learning method for learning similarities and differences. In the method, samples of a same type are aggregated, and samples of different types are distinguished, to learn a similarity representation.

[0024] Large-scale pre-trained language model: It is a pre-trained model obtained through performing pre-training on a large-scale language model (Large Language Model, LLM) on a large-scale corpus, and is configured to perform a natural language processing task. These models each generally include a plurality of layers of neural networks, an input thereof is a text sequence, and an output is a language representation for the text sequence. Pre-trained means that the model has been trained and has pre-learned to process a large amount of language data before a specific task. By pre-training the model, the model can capture more complex language and semantic rules, thereby performing well in various natural language processing tasks and reducing a requirement for large-scale data for specific tasks. A common large-scale pre-trained language model includes, for example, a BERT and a plurality of other LLMs.

[0025] BERT: A full name is Bidirectional Encoder Representations from Transformer. It is a bidirectional encoder representation model based on a Transformer, and is a pre-trained language representation model.

[0026] Visual question answering task: It is to determine, according to an input image and question, an answer to the question from visual information of the input image.

[0027] Image caption task: It is to generate a description text of an input image.

[0028] Visual entailment task: It is to predict semantic relevance, that is, entailment,

neutrality, or contradiction, of an input image and text.

[0029] Referring expression comprehension task: It is to localize, according to input text, an image area corresponding to the input text in an input image.

[0030] Image generation task: It is to generate an image based on an input description text.

[0031] Text-based sentiment classification task: It is to predict sentiment classification information of input text.

[0032] Text summarization task: It is to generate summary information of input text.

[0033] Multi-modal task: It refers to a downstream task, such as a visual question answering task, an image caption task, a visual entailment task, a referring expression comprehension task, or an image generation task, whose input and output data involve a plurality of types of modal data such as images and texts.

[0034] Multi-modal pre-trained model: It refers to a pre-trained model whose input and output data involve a plurality of types of modal data such as images and texts, and may be used in multi-modal task processing after being fine-tuned and trained.

[0035] A large model refers to a deep learning model with large-scale model parameters, and generally includes hundreds of millions, tens of billions, or even hundreds of billions of model parameters. The large model may alternatively be referred to as a foundation model. The large model is pre-trained based on large-scale unannotated corpora, to produce a pre-trained model with more than 100 million parameters. The model, such as a large-scale language model (LLM) or a multi-modal pre-trained model, can adapt to a wide range of downstream tasks and has a good generalization capability. The large model may be widely used in fields such as natural language processing (NLP) and computer vision. Specifically, the large model may be used in computer vision tasks such as visual question answering (VQA), image caption (IC), and image generation, and natural language processing tasks such as text-based sentiment classification, text summary generation, and machine translation. Main application scenarios of the large model include a digital assistant, an intelligent robot, searching, online education, office software, e-commerce, an intelligent design, and the like.

[0036] As a general model, the pre-trained language model has a training target greatly different from a target of the text representation model during a pre-training stage. To improve text representation quality, large-scale high-quality manually annotated data sets are usually used, to perform strong supervision training on the pre-trained language model to improve a text representation capability of the model. However, manual data annotation is time-consuming and labor-intensive and has low annotation consistency, and it is difficult to obtain large-scale high-quality annotated data to train the model. As a result, a text representation

model obtained through training based on the pre-trained language model has a poor text representation capability and low text representation quality, which further leads to a poor effect of a downstream processing task.

[0037] According to a text data processing method provided in the present disclosure, level classification is performed on text data used for training, to obtain a plurality of training sets of different data levels; and multi-stage training is performed on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels, to obtain a text representation model. A data level of the training set represents quality of the training set, that is, quality of training data in the training set. Quality of a training set of a high data level is better than quality of a training set of a low data level, and a better effect is achieved by using the training set of the high data level to perform model training. According to the method in the present disclosure, the text data used for training is classified into the plurality of training sets of different data levels; and multi-stage training is performed on the pre-trained model based on the plurality of training sets of different data levels. Compared with performing one-time (one-stage) training on the pre-trained model based on all training data, this can better improve a representation capability of the pre-trained model, thereby improving a text representation capability of the trained text representation model, improving text representation quality, and further improving an effect of a downstream text processing task.

[0038] In the embodiments of the present disclosure, a pre-trained model used for obtaining a text representation model through training is a machine learning model that can implement text representation, and may be specifically a pre-trained language model such as a BERT, or any other pre-trained large-scale language model LLM, or a model constructed based on a large-scale pre-trained language model. This is not specifically limited herein.

[0039] The text representation model obtained through multi-stage training may be used in any one of the following text-based processing tasks: semantic retrieval, clustering, recommendation, classification, or question answering. In a downstream text processing task, the text representation model is configured to generate a text representation of a given text. Further, the processing task such as semantic retrieval, clustering, recommendation, classification, or question answering is implemented based on the text representation of the given text.

[0040] In addition, the text representation model may alternatively be used in a multi-modal processing task, to implement encoding of text information in the task and obtain a high-quality text representation. Based on the high-quality text representation, combined with a representation of other modal data, the multi-modal processing task may be implemented and

an effect of the multi-modal processing task may be improved.

[0041] In addition, the text representation model may alternatively be used as a foundation model of a downstream task model, and a target task model for executing a downstream task is constructed based on a text representation capability of the foundation model.

[0042] For example, the text representation model is used as a foundation model, and a classifier is superimposed after a last layer of the foundation model, to obtain a text classification model. The text representation model is used as a text representation module in the text classification model. In a text classification task, a text representation of input text is obtained through the text representation module (that is, the text representation model obtained through multi-stage training) in the text classification model, and classification prediction is performed through the classifier based on the text representation of the input text, to obtain a classification result of the input text. In addition, a model used in another downstream task such as smart question answering or a multi-modal task may alternatively be constructed based on the foundation model of the text representation model. This is not specifically limited herein.

[0043] FIG. 1 is a schematic diagram of an exemplary system architecture to which the present disclosure is applicable. As shown in FIG. 1, the system architecture includes a server and an end-side device. There is a functional communication link between the server and the end-side device, which can implement a communication connection between the server and the end-side device.

[0044] The server may be a server cluster deployed on a cloud, or a local device with a computation capability. The server is configured to obtain and store a pre-configured multi-stage training policy and a plurality of training sets of different data levels. The plurality of training sets of different data levels are obtained through performing multi-level classification on text data used for training. The server is further responsible for performing multi-stage training on a pre-trained model by using the configured multi-stage training policy and the plurality of training sets of different data levels, to obtain a text representation model. In addition, the server may be further responsible for constructing large-scale text data used for training, and performing multi-level classification on the text data based on a preset rule, to obtain the plurality of training sets of different data levels. Further, for the obtained text representation model, the server may expose a text representation service based on the obtained text representation model. Alternatively, the server may provide the text representation model or model parameters of the text representation model to the end-side device. The end-side device may obtain the text representation model, or initialize the pre-trained model based on the obtained model parameters of the text representation model, to obtain the text

representation model. A quantity of training sets of different data levels is represented by m in FIG. 1, where m is a positive integer.

[0045] The end-side device is responsible for executing a downstream task based on a text representation capability of the text representation model, for example, executing a text-based downstream task such as semantic retrieval, clustering, recommendation, classification, or question answering. The end-side device may be an electronic device that runs a downstream task system, and may be specifically a hardware device with a network communication function, a calculation function, and an information display function. The end-side device includes, but is not limited to, a smartphone, a tablet computer, a desktop computer, a local server, a cloud server, and the like.

[0046] Based on the system architecture shown in FIG. 1, when needing to obtain a text representation of text information in a process of running a text-based processing task, the end-side device sends an invoking request for the text representation model to the server. The invoking request includes to-be-processed text information. The server receives the invoking request to obtain the to-be-processed text information, inputs the to-be-processed text information into the text representation model for encoding, to obtain a text representation of the text information, and returns the text representation of the text information to the end-side device. The end-side device receives the text representation of the text information returned by the server, and continues executing processing logic of a downstream task according to the text representation of the text information, thereby achieving the downstream processing task.

[0047] In addition, the server may alternatively provide the text representation model (or the model parameter of the text representation model) to the end-side device. The end-side device may locally use the text representation model, and implement, based on the text representation capability of the text representation model, a specific downstream task, for example, a text-based task such as semantic retrieval, clustering, recommendation, classification, or question answering. Certainly, the text representation model may alternatively be used in other tasks in the fields of natural language processing and artificial intelligence. This is not specifically limited herein.

[0048] The following describes the technical solutions of the present disclosure and how to resolve the foregoing technical problems according to the technical solutions of the present disclosure in detail by using specific embodiments. The following several specific embodiments may be combined with each other, and the same or similar concepts or processes may not be described repeatedly in some embodiments. The following describes embodiments of the present disclosure with reference to the accompanying drawings.

[0049] FIG. 2 is a flowchart of a text data processing method according to an exemplary embodiment of the present disclosure. This embodiment is performed by the server in the foregoing system architecture. As shown in FIG. 2, the method specifically includes the following steps.

[0050] Step S201: Perform level classification on text data used for training, to obtain a plurality of training sets of different data levels.

[0051] The text data is pre-obtained training data used for training a text representation capability of a pre-trained model, and includes, but is not limited to, large/super large-scale weakly supervised data and small-scale strongly supervised data. The weakly supervised data may be obtained from a public data set. Alternatively, public text corpora may be collected as much as possible through various proper ways/manners, and a key text may be extracted from the public text corpora to form the text data. For example, two content-related text segments may be extracted to form a text pair as the text data. When the weakly supervised data is constructed, fields, data types, and data sources of the text corpora are not limited, and text corpora of various fields, sources, and data types may be obtained as much as possible, to increase a scale of the weakly supervised data. A manner of obtaining the weakly supervised data is not specifically limited in this embodiment. In addition, in this embodiment, a good training effect may be achieved by using the small-scale strongly supervised data. Therefore, manually annotated text data may be obtained from an existing public data set, to obtain a strongly supervised data set.

[0052] In this embodiment, for large-scale text data used for training, the server performs multi-level classification on the text data, to classify the text data into a plurality of training sets, where different training sets correspond to different data levels. A data level of the training set represents quality of the training set, that is, quality of training data in the training set. Quality of a training set of a high data level is better than quality of a training set of a low data level, and a better effect is achieved by using the training set of the high data level to perform model training.

[0053] In an optional implementation, the step may be specifically implemented in the following manner:

calculating use indicators of the text data according to use behavior data of source data of the text data; and classifying the text data into the plurality of training sets of different data levels according to the use indicators of the text data. A use indicator of text data in a training set of a high data level is greater than a use indicator of text data in a training set of a low data level. For example, for any two training sets of different data levels, if a minimum

value of a use indicator of text data in a training set of a first data level is greater than a maximum value of a use indicator of text data in another training set of a second data level, the first data level is higher than the second data level.

[0054] The source data of the text data refers to a text corpus from which the text data is extracted, and the use behavior data of the source data refers to use behavior data of the text corpus. For different text corpora, data items included in use behavior data may be different. The use behavior data of the text corpus includes, but is not limited to: data of behaviors such as tapping/clicking, browsing, forwarding, collecting, commenting, replying, citing, and being retrieved, or may include other data that can reflect a use condition of the text corpus.

[0055] The use indicator of the text data is an indicator reflecting a use condition of the source data of the text data, and may be calculated according to the use behavior data of the source data of the text data. For a plurality of pieces of use behavior data of different types of a piece of source data, different statistical indicators may be respectively calculated as use indicators. For example, a click-through rate, number of views, number of forwarding, number of collections, number of comments, number of citations, and number of retrievals of the source data of the text data may be calculated. For a plurality of use indicators of a piece of text data, a weight system may be set for different types of use indicators, and weighted summation/weighted averaging is performed on the plurality of use indicators, to obtain a comprehensive indicator.

[0056] Optionally, according to a quantity of pre-configured data levels and an indicator range corresponding to each data level, the text data is classified into a training set of a corresponding data level according to within which indicator range of the corresponding data level a comprehensive indicator of the text data falls.

[0057] Optionally, the text data may alternatively be sorted according to comprehensive indicators of all text data, the text data may be classified into a plurality of different training sets according to a sorting result, and then data levels of the training sets may be sequentially set. For example, according to the sorting result, the text data is equally classified or classified into a plurality of parts according to a specific ratio, to form the plurality of training sets.

[0058] In this implementation, the server classifies, based on the use indicators of the text data, the text data into the plurality of training sets of different data levels according to a pre-configured classification rule. Specifically, the classification rule may be configured and adjusted according to an actual requirement. This is not specifically limited herein.

[0059] In another optional implementation, the step may be specifically implemented in the following manner:

classifying the text data into training sets of corresponding data levels according to data types of the source data of the text data and a pre-configured data level corresponding to each data type. For example, text data extracted from knowledge graphs, academic literature, and question answering data may be classified into a high-level training set; and text data extracted from online articles and comment data may be classified into a low-level training set. In addition, the data levels corresponding to the data types may be configured and adjusted according to an actual requirement. This is not specifically limited herein.

[0060] In another optional implementation, the step may be specifically implemented in the following manner:

according to whether the text data is annotated, classifying unannotated text data into a weakly supervised training set, and classifying annotated text data into a strongly supervised training set, where a data level of the weakly supervised training set is lower than a data level of the strongly supervised training set.

[0061] In actual application, annotated training data is high-quality strongly supervised data, and unannotated training data is generally low-quality weakly supervised data. In this implementation, the text data may be naturally classified into two training sets according to whether the text data is annotated, where the annotated text data is in the strongly supervised training set, and the unannotated text data is in the weakly supervised training set. A specific annotation manner of the text data may be configured according to a requirement of an actual application scenario. This is not specifically limited herein.

[0062] In another optional implementation, when the text representation capability of the pre-trained model is trained and optimized, the text data used for training may include a plurality of text pairs, and the text pair includes two content-related texts. Contrastive learning training is performed on the pre-trained model by using the text pair, which can well improve the text representation capability of the pre-trained model. This step may be specifically implemented in the following manner: classifying the text data into the plurality of training sets of different data levels according to correlations of the text pairs, where a correlation of a text pair in a training set of a high data level is greater than a correlation of a text pair in a training set of a low data level.

[0063] The correlations of the text pairs may be obtained by using any existing text relevance analysis algorithm, such as TF-IDF (Term Frequency-Inverse Document Frequency) algorithm or a text matching algorithm. This is not specifically limited herein.

[0064] Optionally, according to a quantity of pre-configured data levels and a correlation range corresponding to each data level, the text pair is classified into a training set of a

corresponding data level according to within which correlation range of the corresponding data level a correlation of the text pair falls. Optionally, the text pairs may alternatively be sorted according to correlations of the text pairs, the text pairs may be classified into a plurality of different training sets according to a sorting result, and then data levels of the training sets may be sequentially set. For example, according to the sorting result, the text pairs are equally classified or classified into a plurality of parts according to a specific ratio, to form the plurality of training sets.

[0065] In this implementation, based on the correlations of the text pairs, the server classifies the text pairs into the plurality of training sets of different data levels according to a pre-configured classification rule. Specifically, the classification rule may be configured and adjusted according to an actual requirement. This is not specifically limited herein.

[0066] In this embodiment, a classification policy used in performing level classification on the text data may be configured and adjusted according to an actual requirement, and is not limited to the classification policies mentioned in the foregoing implementations. This is not specifically limited herein.

[0067] Step S202: Perform multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels, to obtain a text representation model, where the text representation model is used for generating a text representation of input text.

[0068] A pre-trained model used for obtaining a text representation model through training is a machine learning model that can implement text representation, and may be specifically a pre-trained language model such as a BERT, or any other pre-trained large-scale language model LLM, or a model constructed based on a large-scale pre-trained language model. This is not specifically limited herein.

[0069] Exemplarily, the pre-trained model may be a pre-trained BERT. When performing text representation, a text is input into the BERT for encoding, and a first token position (that is, a CLS position) output by a last layer of the BERT is used as a text representation of the input text.

[0070] Exemplarily, the pre-trained model may be obtained through superimposition of a mean pooling layer after a transformer structure's output layer of a last layer of a pre-trained language model. When performing text representation, the text is input into the pre-trained language model, and the text representation of the input text is obtained after a mean pooling operation performed on a representation of the transformer structure's output layer of the last layer of the pre-trained language model.

[0071] In this embodiment, a training manner of performing multi-stage training on the pre-trained model based on the plurality of training sets of different data levels is configured in the multi-stage training policy.

[0072] Optionally, a quantity of training stages may not be limited in the multi-stage training policy, and according to the classified training sets of different data levels, stage training is performed on the pre-trained model by sequentially using the training sets in ascending order of data level, to obtain the text representation model. First, the pre-trained model is trained by using large-scale low-level text data, which can gradually improve the text representation capability and a characteristic of the pre-trained model. Subsequently, the pre-trained model is more accurately trained by using high-level text data, which can further improve the text representation capability of the pre-trained model, to obtain a stronger text representation model.

[0073] Exemplarily, the text data is classified into the large/super large-scale weakly supervised training set and the small-scale strongly supervised training set, and first the large/super large-scale weakly supervised training set is used to perform weakly supervised contrastive learning training on the pre-trained model, to obtain a first-stage model, to significantly improve the text representation capability and the characteristic of the pre-trained model. Then the small-scale strongly supervised training set is used to perform strongly supervised contrastive learning training on the first-stage model, to further improve the text representation capability of the model, to obtain a text representation model with a strong text representation capability.

[0074] Optionally, the multi-stage training policy includes a configured quantity of training stages and training sets used in the training stages. In the configured multi-stage training policy, a quantity of the training sets of different data levels may be or may not be equal to the quantity of training stages. One or more training sets may be used for training in one training stage, and one training set is used in zero, one, or more training stages. In this step, the pre-trained model is trained by using the configured multi-stage training policy and sequentially using the training sets corresponding to the training stages, to obtain the text representation model.

[0075] Exemplarily, it is assumed that the text data is classified into training sets of five data levels, which are S1, S2, S3, S4, and S5 from low to high. According to a requirement of actual application, three training stages may be configured in the multi-stage training policy. S1 and S2 are used in a first stage, S2, S3, and S4 are used in a second stage, and S5 is used in a third stage.

[0076] In this embodiment, the multi-stage training policy may be configured and adjusted

according to a requirement of an actual application scenario and a data characteristic of a training set that is actually obtained. This is not specifically limited in this embodiment.

[0077] In an optional embodiment, the multi-stage training policy may be customized or updated by a user who obtains the text representation model. Specifically, in response to a configuration/update request of the multi-stage training policy, the server outputs the multi-stage training policy, data levels of the training sets, and description information of the training sets through a training policy configuration interface. The description information of the training set may include, but is not limited to relevant information such as source information, a field, a data composition structure/format, an applicable scenario, and a data feature of data in a data set, which can guide the user to use the training set more properly and correctly. The user may learn a characteristic and a data level of each training set through the description information of the training set displayed on the training policy configuration interface, and customize or update the multi-stage training policy.

[0078] Further, the server configures/updates the multi-stage training policy in response to an editing operation on any configuration information in the multi-stage training policy. The configuration information includes: a quantity of training stages, and training sets used in the training stages. The user may modify the quantity of training stages in the multi-stage training policy, or may adjust a training set specifically used in any training stage.

[0079] In an optional embodiment, the server may output a current training result through a front-end interface after performing any stage of training on the pre-trained model. The current training result is used for guiding adjustment of the multi-stage training policy. The training result may be information, such as various performance indicators (such as an accuracy rate and a recall rate) of the model, and a stability indicator of model performance (such as a degree of performance improvement/degradation), that reflects a current training effect. Through the current training result output through the front-end interface, the user may determine, according to an actual application requirement, whether to continue training, adjust the multi-stage training policy and continue training, stop training, or the like.

[0080] Further, after the user updates the multi-stage training policy, in response to the update of the multi-stage training policy, the server uses the updated multi-stage training policy to perform a subsequent stage of training on the pre-trained model. Adjusting the multi-stage training policy in time is supported, which improves a training effect of the text representation model, thereby improving the text representation capability of the text representation model.

[0081] According to the method in this embodiment, the text data used for training is classified into the plurality of training sets of different data levels; and multi-stage training is

performed on the pre-trained model based on the plurality of training sets of different data levels. Compared with performing one-time (one-stage) training on the pre-trained model based on all training data, this can better improve a representation capability of the pre-trained model, thereby improving a text representation capability of the trained text representation model, improving text representation quality, and further improving an effect of a downstream text processing task.

[0082] In an optional embodiment, after the text representation model is obtained by performing multi-stage training on the pre-trained model by using the plurality of training sets of different data levels, the text representation model may be further fine-tuned and trained by using a dedicated training set of a vertical field, to obtain a text representation model tailored for the vertical field.

[0083] The dedicated training set of the vertical field may be constructed by the user according to a vertical field in which the text representation model is actually used and data of an actual task scenario.

[0084] Exemplarily, the user constructs the dedicated training set of the vertical field through an end-side device, and submits the dedicated training set of the vertical field to the server. The server receives the dedicated training set of the vertical field submitted by the end-side device, and uses the dedicated training set to train the text representation model, to obtain the text representation model tailored for the vertical field. Further, the server provides the text representation model tailored for the vertical field for the end-side device.

[0085] For example, the server provides a download address of the text representation model tailored for the vertical field for the end-side device. The end-side device downloads, according to the download address, the text representation model tailored for the vertical field, and locally runs the text representation model tailored for the vertical field. For example, the server sends a model parameter of the text representation model tailored for the vertical field to the end-side device. The end-side device uses the received model parameter to initialize a local text representation model, to obtain the text representation model tailored for the vertical field. For example, the server provides an application programming interface (API) of the text representation model of the vertical field for the end-side device, and when needing to use the text representation model, the end-side device uses, by invoking the API of the text representation model, the text representation model to encode input text, to obtain a text representation of the text.

[0086] In another typical embodiment, the text data used for training includes a plurality of text pairs, where the text pair includes two content-related texts. According to whether the

text pair is annotated, an unannotated text pair is classified into a weakly supervised training set, and an annotated strongly related text pair is classified into a strongly supervised training set. When annotation of the text pair is performed, whether there is strong relevance between content in the text pair is annotated, an irrelated text pair is removed, and a strongly related text pair is retained. The weakly supervised training set includes large/super large-scale unannotated text pairs, and the strongly supervised training set includes small-scale annotated strongly related text pairs. A data level of the weakly supervised training set is lower than a data level of the strongly supervised training set.

[0087] FIG. 3 is a framework diagram of a text data processing method according to an embodiment. In this embodiment, as shown in FIG. 3, used training data includes a large-scale weakly supervised training set and a small-scale strongly supervised training set. When multi-stage training is performed, in first-stage training, the large-scale weakly supervised training set is used to perform weakly supervised contrastive learning on a pre-trained model, to improve a text representation capability and a field generalization capability of the pre-trained model, to obtain a first-stage model. Further, in second-stage training, initialization is performed by using the first-stage model, and the strongly supervised training set is used to perform strongly supervised contrastive learning on the first-stage model, to further enhance the text representation capability of the model. After the training is completed, a model with a strong text representation capability is obtained as a text representation model, which may greatly improve a text representation capability of the text representation model and improve text representation quality.

[0088] FIG. 4 is a flowchart of a text data processing method according to an embodiment of the present disclosure. This embodiment is performed by the server in the foregoing system architecture. As shown in FIG. 4, the method specifically includes the following steps.

[0089] Step S401: Obtain a pre-trained model used for implementing text representation.

[0090] In this embodiment, a pre-trained model used for obtaining a text representation model through training is a machine learning model that can implement text representation, and may be specifically a large-scale pre-trained language model such as a BERT, or any other pre-trained large-scale language model LLM, or a model constructed based on a large-scale pre-trained language model. This is not specifically limited herein.

[0091] Exemplarily, the pre-trained model may be a pre-trained BERT. When performing text representation, a text is input into the BERT for encoding, and a first token position (that is, a CLS position) output by a last layer of the BERT is used as a text representation of the input text.

[0092] Exemplarily, the pre-trained model may be obtained through superimposition of a mean pooling layer after a transformer structure's output layer of a last layer of a pre-trained language model. When performing text representation, the text is input into the pre-trained language model, and the text representation of the input text is obtained after a mean pooling operation performed on a representation of the transformer structure's output layer of the last layer of the pre-trained language model.

[0093] In this embodiment, the pre-trained model used for implementing text representation is constructed based on an existing large-scale language model, a pre-trained model, or the like, and may be specifically implemented in a manner of constructing the text representation model in the related art. The pre-trained model may be a to-be-trained text representation model of any structure in the related art. This is not specifically limited herein.

[0094] In this embodiment, a large/super large-scale weakly supervised data set and a small-scale strongly supervised data set are pre-constructed. The weakly supervised training set may be obtained from similar public data sets. Alternatively, public text corpora may be collected as much as possible through various proper ways/manners, and two content-related text segments may be extracted from the text corpora to form a text pair. A manner of obtaining the weakly supervised training set is not specifically limited in this embodiment. When the weakly supervised training set is constructed, fields and data types of the text corpora are not limited, and text corpora of various fields, sources, and data types may be obtained as much as possible.

[0095] The strongly supervised training set (that is, a supervised data set) includes a plurality of text pairs, and one text pair includes two related texts. Relevance between text pairs included in the strongly supervised training set is manually annotated, and there is strong relevance between content in the text pairs. In this embodiment, a good training effect can be achieved by using a small-scale strongly supervised training set. Therefore, manually annotated text pairs may be obtained from existing public data sets, so that the strongly supervised training set may be obtained.

[0096] In this embodiment, the text pair includes: a first text and a second text related to the first text. Any text pair may be represented as (s_i, s_i^+), where s_i represents the first text, and s_i^+ represents the second text related to s_i in content.

[0097] Two-stage contrastive learning training is performed in a process of training the pre-trained model to obtain the text representation model. In first-stage training, the large/super large-scale weakly supervised training set is used to perform weakly supervised contrastive

learning training on the pre-trained model, to obtain a first-stage model, which may effectively improve a text representation capability and a field generalization capability of the model. In second-stage training, the strongly supervised training set is used to perform strongly supervised contrastive learning training on the first-stage model, to obtain the text representation model, which may further improve the text representation capability of the model. Obtaining a text representation model with a good text representation capability and a good characteristic may improve quality of text representation.

[0098] Step S402: Perform contrastive learning training on the pre-trained model by using the weakly supervised training set, to obtain the first-stage model.

[0099] In this step, first-stage training is performed on the pre-trained model, and contrastive learning training is performed on the pre-trained model by using text pairs on which relevance annotation is not performed and that are included in the weakly supervised training set, to implement weakly supervised contrastive learning training, to obtain the first-stage model.

[0100] Specifically, when contrastive learning training is performed on the pre-trained model by using the text pairs on which relevance annotation is not performed, a plurality of times of iterative training are performed on the pre-trained model based on the text pair included in the weakly supervised training set. During each iteration, a batch of text pairs are sampled, and optimization of a model parameter is performed based on the sampled batch of text pairs. When an iteration stop condition is satisfied, the first-stage training is completed, and the first-stage model is obtained.

[0101] A batch size of the sampled text pairs in each iteration may be expressed as K. A value of K may be set according to an actual application scenario and experience. For example, K may be set to 128, 300, or 512. This is not specifically limited in this embodiment. In addition, the iteration stop condition may be that a quantity of iterations reaches a preset quantity-of-times threshold, or a model effect is stable. Details may be set according to an actual application scenario and experience. This is not specifically limited in this embodiment.

[0102] In an optional implementation, in an iterative training process, for a first text s_i in any text pair (s_i, s_i^+), a second text s_i^+ belonging to a same text pair as the first text s_i is used as a positive sample s_i^+ relevant to the first text s_i, and second texts belonging to different text pairs from the first text s_i are used as negative samples (represented as $s_{i,1}^-$, $s_{i,2}^-$, ...) irrelevant to the first text s_i. Contrastive learning training is performed on the pre-trained model based on the constructed positive sample s_i^+ and the constructed negative samples

$s_{i,1}^{-}$, $s_{i,2}^{-}$, ... of the first text s_i, to obtain the first-stage model.

[0103] For example, a sampling batch size is 128. For any text pair (s_i, s_i^{+}) in sampled 128 text pairs, for a first text s_i, there is only one second text belonging to a same text pair as the first text s_i, and there is only one positive sample, that is, s_i^{+}, of the first text s_i. There are 127 second texts belonging to different text pairs from the first text s_i, and there are 127 negative samples of the first text s_i, including: $s_{i,1}^{-}$, $s_{i,2}^{-}$, ... , $s_{i,127}^{-}$.

[0104] In another optional implementation, in an iterative training process, for a first text s_i in any text pair (s_i, s_i^{+}), a second text s_i^{+} belonging to a same text pair as the first text s_i is used as a positive sample s_i^{+} relevant to the first text s_i. A first text that belongs to a different text pair from the first text s_i and a second text that belongs to a different text pair from the first text s_i are all used as negative samples (represented as $s_{i,1}^{-}$, $s_{i,2}^{-}$, ...) irrelevant to the first text s_i. Contrastive learning training is performed on the pre-trained model based on the constructed positive sample s_i^{+} and the constructed negative samples $s_{i,1}^{-}$, $s_{i,2}^{-}$, ... of the first text s_i, to obtain the first-stage model.

[0105] For example, a sampling batch size is 128. For any text pair (s_i, s_i^{+}) in sampled 128 text pairs, for a first text s_i, there is only one second text belonging to a same text pair as the first text s_i, and there is only one positive sample, that is, s_i^{+}, of the first text s_i. There are 254 (127×2) first texts and second texts that belong to different text pairs from the first text s_i. Therefore, there are 254 negative samples of the first text s_i, including: $s_{i,1}^{-}$, $s_{i,2}^{-}$, ... , $s_{i,254}^{-}$.

[0106] In another optional implementation, in an iterative training process, for a first text s_i in any text pair (s_i, s_i^{+}), a second text s_i^{+} belonging to a same text pair as the first text s_i is used as a positive sample s_i^{+} relevant to the first text s_i. However, a negative sample of the first text s_i may be obtained by sampling a specified quantity of texts from first texts and/or second texts of other text pairs that do not include the first text s_i. For a first text, a specified quantity of sampled negative samples may be set according to an actual application scenario. This is not specifically limited herein.

[0107] Further, when iterative training is performed, two texts of the text pair included in the weakly supervised training set are separately input into the pre-trained model, and text representations of the two texts of the text pair are generated through the pre-trained model, so that a text representation of any text included in the weakly supervised training set may be obtained.

[0108] Optionally, when the text representations of the two texts of the text pair are generated by using the pre-trained model, a dual-tower model may be further constructed based

on the pre-trained model. The dual-tower model includes two parallel branches, and the two branches use the same pre-trained model. The two texts of the text pair are respectively input into the two branches of the dual-tower model, and the text representations of the two texts may be obtained.

[0109] Further, for a first text s_i in any text pair, based on a constructed positive sample s_i^+ and constructed negative samples $s_{i,1}^-$, $s_{i,2}^-$, ... of the first text s_i, a correlation between the first text s_i and the positive sample s_i^+ is calculated according to a text representation of the first text s_i and a text representation of the positive sample s_i^+, and may be represented as $\mathrm{sim}(s_i, s_i^+)$. According to the text representation of the first text s_i and a text representation of any negative sample $s_{i,j}^-$, a correlation between the first text s_i and the negative sample $s_{i,j}^-$ is calculated, and may be represented as $\mathrm{sim}(s_i, s_{i,j}^-)$. A contrastive learning loss function is calculated according to the correlation $\mathrm{sim}(s_i, s_i^+)$ between the first text s_i and the positive sample s_i^+ and the correlation $\mathrm{sim}(s_i, s_{i,j}^-)$ between the first text s_i and the negative sample $s_{i,j}^-$. Taking the contrastive learning loss function as an optimization target, based on the contrastive learning loss function, a parameter of the pre-trained model is trained, to increase the correlation $\mathrm{sim}(s_i, s_i^+)$ between the first text s_i and the positive sample s_i^+, and reduce the correlation $\mathrm{sim}(s_i, s_{i,j}^-)$ between the first text s_i and the negative sample $s_{i,j}^-$, thereby implementing contrastive learning training of the pre-trained model and improving the text representation capability and the characteristic of the pre-trained model.

[0110] Exemplarily, the contrastive learning loss function may be represented as:

$$L = \sum_i^K L\left(s_i, s_i^+, s_{i,1}^-, s_{i,2}^-, \dots, s_{i,n}^-\right) = \sum_i^K \frac{e^{\mathrm{sim}(s_i, s_i^+)}}{e^{\mathrm{sim}(s_i, s_i^+)} + \sum_{j=1}^n e^{\mathrm{sim}(s_i, s_{i,j}^-)}},$$

where n represents a quantity of negative samples of the first text s_i, and e is a natural constant. $\mathrm{sim}()$ represents a correlation between the two texts, and specifically may be a cosine similarity of the text representations of the two texts.

[0111] Exemplarily, the contrastive learning loss function is also represented as:

$$L = \sum_i^K L\left(s_i, s_i^+, s_{i,1}^-, s_{i,2}^-, \dots, s_{i,n}^-\right) = \sum_i^K \frac{\mathrm{sim}(s_i, s_i^+)}{\mathrm{sim}(s_i, s_i^+) + \sum_{j=1}^n \mathrm{sim}(s_i, s_{i,j}^-)},$$

or $L = \sum_i^K L\left(s_i, s_i^+, s_{i,1}^-, s_{i,2}^-, \dots, s_{i,n}^-\right) = -\sum_i^K log \frac{e^{\mathrm{sim}(s_i, s_i^+)}}{e^{\mathrm{sim}(s_i, s_i^+)} + \sum_{j=1}^n e^{\mathrm{sim}(s_i, s_{i,j}^-)}}$, where log is a logarithmic function.

[0112] In this embodiment, the contrastive learning loss function may alternatively be

obtained based on a change of $L = \sum_i^K L\left(s_i, s_i^+, s_{i,1}^-, s_{i,2}^-, \ldots, s_{i,n}^-\right) = \sum_i^K \frac{\mathrm{sim}(s_i, s_i^+)}{\mathrm{sim}(s_i, s_i^+) + \sum_{j=1}^n \mathrm{sim}(s_i, s_{i,j}^-)}$, and specifically may be set according to an actual requirement and experience. This is not specifically limited herein.

[0113] After the plurality of times of iterative training, when the iteration stop condition is satisfied, the first-stage training is completed, and the first-stage model is obtained.

[0114] Step S403: Perform contrastive learning training on the first-stage model by using the strongly supervised training set, to obtain the text representation model.

[0115] In this step, second-stage training is performed on the first-stage model, and contrastive learning training is performed on the first-stage model by using text pairs on which relevance annotation is performed and that are included in the strongly supervised training set, to implement strongly supervised contrastive learning training, to obtain the text representation model.

[0116] Specifically, when contrastive learning training is performed on the first-stage model by using the text pairs included in the strongly supervised training set, a plurality of times of iterative training are performed on the first-stage model based on the text pairs. During each iteration, a batch of text pairs are sampled, and optimization of a model parameter is performed based on the sampled batch of text pairs. When an iteration stop condition is satisfied, the second-stage training is completed, and the text representation model is obtained.

[0117] In an optional embodiment, for a first text in any text pair in the strongly supervised training set, a second text belonging to a same text pair as the first text is used as a positive sample relevant to the first text, and a second text belonging to a different text pair from the first text is used as a negative sample irrelevant to the first text. Contrastive learning training is performed on the first-stage model, to obtain the text representation model.

[0118] In another optional embodiment, for a first text in any text pair in the strongly supervised training set, a second text belonging to a same text pair as the first text is used as a positive sample relevant to the first text. A first text that belongs to a different text pair from the first text and a second text that belongs to a different text pair from the first text are both used as negative samples irrelevant to the first text. Contrastive learning training is performed on the first-stage model, to obtain the text representation model.

[0119] In another optional embodiment, for a first text in any text pair in the strongly supervised training set, a second text belonging to a same text pair as the first text is used as a positive sample relevant to the first text. However, a negative sample of the first text may be obtained by sampling a specified quantity of texts from first texts and/or second texts of other

text pairs that do not include the first text. Contrastive learning training is performed on the first-stage model, to obtain the text representation model. For a first text, a specified quantity of sampled negative samples may be set according to an actual application scenario. This is not specifically limited herein.

[0120] In this step, a process of contrastive learning training in the second stage is similar to contrastive learning training in the foregoing first stage, and a difference is that used training sets and models for initialization are different. In the first stage, the weakly supervised training set is used, the pre-trained model is used for initialization, and the pre-trained model is trained by using the weakly supervised training set. In the second stage, the strongly supervised training set is used, the first-stage model obtained after the first-stage training is used for initialization, and the first-stage model is trained by using the strongly supervised training set. A specific training process of the second stage is not described again herein in this embodiment, and reference may be made to related descriptions of the training process of the first stage.

[0121] It should be noted that, the training process of the first stage is similar to the training process of the second stage. However, considering that sizes of data sets used in the first stage and the second stage are different, different batch sizes and iteration stop conditions may be configured for the two stages, and different contrastive loss functions may also be configured for the two stages, to improve training effects of the two stages and obtain a text representation model with a good text representation capability and a good characteristic.

[0122] After the second-stage training is completed, the text representation model may be obtained. Further, for the obtained text representation model, the server may expose a text representation service based on the obtained text representation model. Based on the system architecture shown in FIG. 1, when needing to obtain a text representation of text information in a process of running a text-based processing task, the end-side device sends an invoking request for the text representation model to the server. The invoking request includes to-be-processed text information. The server receives the invoking request to obtain the to-be-processed text information, inputs the to-be-processed text information into the text representation model for encoding, to obtain a text representation of the text information, and returns the text representation of the text information to the end-side device. The end-side device receives the text representation of the text information returned by the server, and continues executing processing logic of a downstream task according to the text representation of the text information, thereby achieving the downstream processing task.

[0123] In addition, the server may further provide the text representation model for the end-side device. The end-side device may locally use the text representation model, and implement,

based on a text representation capability of the text representation model, a specific downstream task, for example, a text-based task such as semantic retrieval, clustering, recommendation, classification, or question answering.

[0124] Exemplarily, the server provides download address information of the text representation model for the end-side device. The end-side device may download, according to the download address information, the text representation model from the server. Alternatively, the server may deliver model parameters of the text representation model to the end-side device. The end-side device initializes the text representation model according to the received model parameters of the text representation model, to obtain a text representation model with a strong text representation capability. Further, the end-side device may locally run the text representation model, and implement a specific downstream task based on the strong text representation capability of the text representation model.

[0125] According to the method in the present disclosure, large/super large-scale text pairs on which relevance annotation is not performed are constructed as the weakly supervised training set, and small-scale text pairs on which relevance annotation is performed are obtained as the strongly supervised training set. When the text representation model is obtained, in first-stage training, the weakly supervised training set is used to perform weakly supervised contrastive learning on the pre-trained model, to improve a text representation capability and a field generalization capability of the pre-trained model, to obtain the first-stage model. Further, in second-stage training, initialization is performed by using the first-stage model, and the strongly supervised training set is used to perform strongly supervised contrastive learning on the first-stage model, to further improve the text representation capability of the model. After the training is completed, a model with a strong text representation capability is obtained as a text representation model, which may greatly improve the text representation capability of the text representation model and improve text representation quality, thereby improving an effect of a downstream text processing task.

[0126] FIG. 5 is a flowchart of constructing a weakly supervised training set according to an exemplary embodiment of the present disclosure. Based on the foregoing embodiments, a process of constructing a large/super large-scale weakly supervised training set is described in detail in this embodiment. As shown in FIG. 5, a specific procedure of constructing a weakly supervised training set including text pairs on which relevance annotation is not performed is as follows.

[0127] Step S501: Collect a plurality of text corpora of different data types.

[0128] When the weakly supervised training set is constructed, the plurality of text corpora

of different data types are collected first, and content-related text pairs are extracted from the text corpora.

[0129] When the text corpora are collected, fields, data types, data sources of the text corpora are not limited, and a plurality of public text corpora of different fields, data sources, and data types are collected as much as possible through proper ways and manners, to obtain a larger scale of text corpora and expand a scope of a knowledge field covered by the text corpora.

[0130] When collecting the text corpus, a server may collect a public data set, web page data, encyclopedia data, community question answering data, and the like. Data produced or constructed by another application platform/system, such as a constructed knowledge graph, generated comment data, question answering data, a retrieval record, and open source code, may also be obtained. The server may alternatively use any existing manner of collecting the text corpora to collect more text corpora. A collection manner and data source of text corpora are not specifically limited herein.

[0131] Obtained text corpora of different data sources may have different data types, including but not limited to an article-type, a question answering-type, comment data, a knowledge graph, code data, and a text link.

[0132] Step S502: Extract, according to the data types of the text corpora, text pairs from the text corpora as text data used for training.

[0133] In this embodiment, different text pair extraction methods may be used for extracting the text pairs from the text corpora of different data types.

[0134] Exemplarily, a collected article-type text corpus generally includes a title, an abstract, a summary, main content, and the like of an article. A topic and a content text in a same text corpus may be extracted from the article-type text corpus; and the topic and the content text in the same text corpus are respectively used as a first text and a second text, to generate a text pair. The topic of the article-type text corpus may be the title, the abstract, the summary, or the like, and the content text refers to all or a segment of the main content. In addition, for a text corpus in which main content of an article is long, the main content of the article may further be divided into a plurality of content segments, and a title (or an abstract, or a summary) of the article and each content segment of the main content of the article form a text pair.

[0135] Exemplarily, for a collected question answering-type text corpus, a question and a response text of the question in a same text corpus are extracted; and the question and the response text of the question are respectively used as a first text and a second text, to generate a text pair. The question answering-type text corpus generally has a structure like: input

(query)-response. Based on this structure, the input (query) and the response are respectively used as the first text and the second text, to construct the text pair.

[0136] Exemplarily, collected comment data generally includes a published topic text (such as an original text or an excerpted text) and a comment on the topic text. Each topic text is related to content of each comment of the topic text, and an originally published topic text and a comment text of the topic text in the comment data may be extracted. The topic text and the comment text of the topic text are respectively used as a first text and a second text, to generate a text pair.

[0137] Exemplarily, a collected knowledge graph generally includes an entity and description information of the entity. There is certain relevance between a name of each entity and description information of the entity, and an entity name and entity description of the entity included in the knowledge graph may be obtained. An entity name and entity description of a same entity are respectively used as a first text and a second text, to generate a text pair.

[0138] Exemplarily, collected code data includes code and an annotation text of the code. The annotation text is related to the corresponding code in content, and the annotation text and the code corresponding to the annotation text in the code data may be obtained. The annotation text and the code corresponding to the annotation text are respectively used as a first text and a second text, to generate a text pair.

[0139] Exemplarily, data of a text link type refers to that a text is given as a quotation (or citation), and the quotation is linked to another longer main text (such as a document). For the data of the text link type, the quotation (or citation) and the main text to which the quotation is linked are extracted, to form a text pair. In addition, the quotation (or citation) and each of a plurality of segments of the main text to which the quotation is linked may alternatively form a text pair.

[0140] In this embodiment, the plurality of text corpora of different data types are collected, and the text pairs are extracted from the text corpora, so that large-scale unannotated text pairs may be constructed, to form the large-scale weakly supervised training set. The weakly supervised training set not only includes the large-scale text pairs, but also covers a wide range of field knowledge. Through the method in this embodiment, a super large-scale weakly supervised training set including more than one billion text pairs may be constructed. Weakly supervised contrastive learning training is performed on the pre-trained model based on the large-scale weakly supervised training set, which may effectively improve a text representation capability and a characteristic of the model, and improve a generalization capability of the model in each field.

[0141] In any method embodiment, a used strongly supervised training set may be obtained by obtaining a public supervised data set. In addition, alternatively, the server may construct the strongly supervised training set. In a process of constructing the strongly supervised training set, a manner of obtaining the text pairs is similar to a manner of constructing the weakly supervised training set, relevance of the text pair is further annotated, and an irrelated text pair or a text pair with weak relevance is removed, so that text pairs with strong relevance on which relevance annotation is performed may be obtained, thereby obtaining the strongly supervised training set. This embodiment only requires a small-scale strongly supervised training set to achieve a good training effect, so that an amount of annotated data can be greatly reduced, thereby reducing time and labor cost of data annotation.

[0142] It should be noted that, to reduce difficulty of manual annotation, a text pair (including question-response) extracted based on question answering-type data may be annotated, which can reduce the difficulty of manual annotation and improve efficiency of annotation to a certain extent.

[0143] In the following embodiments, application of the text representation model that is obtained by performing multi-stage training on the pre-trained model by using the configured multi-stage training policy and using the plurality of training sets of different data levels is exemplarily described.

[0144] FIG. 6 is a flowchart of a text representation method according to an exemplary embodiment of the present disclosure. The server in the foregoing system architecture may locally run the text representation model or deploy the text representation model to another server after obtaining the text representation model. This embodiment is performed by a server running the text representation model. As shown in FIG. 6, the method specifically includes the following steps.

[0145] Step S601: Receive an invoking request from an end-side device for the text representation model, where the invoking request includes to-be-processed text information.

[0146] Exemplarily, the server may expose an application programming interface (API) of the text representation model. The end-side device may invoke the text representation model through the API.

[0147] When the end-side device invokes the API of the text representation model, the server extracts the to-be-processed text information from the invoking request after receiving the invoking request from the end-side device for the text representation model.

[0148] Step S602: Input the text information into the text representation model for encoding, to obtain a text representation of the text information, where the text representation

model is obtained through performing level classification on text data used for training to obtain a plurality of training sets of different data levels, and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels.

[0149] After the to-be-processed text information is obtained, the text information is input into the text representation model, and the text information is encoded into the text representation through the text representation model.

[0150] The text representation model used in this embodiment is obtained through the method of the foregoing method embodiments: performing level classification on text data used for training, to obtain a plurality of training sets of different data levels; and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels. The text representation model has a good text representation capability and a good characteristic, which can improve quality of text representation. For a specific implementation and technical effect of obtaining the text representation model, refer to related content in the foregoing embodiments, and details are not described herein again.

[0151] Step S603: Output the text representation of the text information to the end-side device.

[0152] After obtaining the text representation of the text information through encoding through the text representation model, the server returns the text representation of the text information to the end-side device.

[0153] In this embodiment, the server running the text representation model exposes an invoking interface of the text representation model. When needing to obtain the text representation of the text information in a process of implementing various text-based processing tasks, the end-side device may invoke the text representation model through the interface, and use the text representation capability of the text representation model.

[0154] The method in this embodiment may be applied to various downstream tasks implemented based on the text representation capability of the text representation model, for example, text-based tasks such as semantic retrieval, clustering, recommendation, classification, and question answering. Certainly, the text representation model may alternatively be used in other tasks in natural language processing and artificial intelligence. This is not specifically limited herein.

[0155] In other embodiments, the server in the foregoing system architecture may further provide the text representation model to the end-side device after obtaining the text

representation model. The end-side device may locally use the text representation model, and implement, based on a text representation capability of the text representation model, a specific downstream task, for example, a text-based task such as semantic retrieval, clustering, recommendation, classification, or question answering. Certainly, the text representation model may alternatively be used in other tasks in natural language processing and artificial intelligence. This is not specifically limited herein.

[0156] FIG. 7 is a flowchart of a text clustering method according to an exemplary embodiment of the present disclosure. This embodiment may be performed by an end-side device or a server with a text representation model running therein. As shown in FIG. 7, the method specifically includes the following steps.

[0157] Step S701: Obtain a plurality of to-be-clustered text segments.

[0158] When applied to different clustering scenarios, according to an object that actually needs to be clustered, a text relevant to the object may be used as the to-be-clustered text segment, and may be specifically a description text of the object, a key text included in the object, or the like. This is not specifically limited herein.

[0159] Exemplarily, that articles published on an application platform need to be clustered is used as an example. Titles of the articles may be used as to-be-clustered text segments, and the titles of the articles are clustered, to cluster titles with similar content together, thereby clustering articles with the titles with similar content together.

[0160] Exemplarily, news analysis is used as an example. News is clustered by topic, so that social hot news can be better understood. Topics of the news are used as to-be-clustered text segments, and the topics of the news are clustered, to cluster news with similar topics together.

[0161] Step S702: Separately input the plurality of text segments into a text representation model for encoding, to generate text representations of the text segments, where the text representation model is obtained through performing level classification on text data used for training to obtain a plurality of training sets of different data levels, and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels.

[0162] After the plurality of to-be-clustered text segments are obtained, the text segments are separately input into the text representation model, and the input text segments are encoded into the text representations through the text representation model, that is, the text representations of the text segments can be obtained.

[0163] The text representation model used in this embodiment is obtained through the

method of the foregoing method embodiments: performing level classification on text data used for training, to obtain a plurality of training sets of different data levels; and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels. The text representation model has a good text representation capability and a good characteristic, which can improve quality of text representation. For a specific implementation and technical effect of obtaining the text representation model, refer to related content in the foregoing embodiments, and details are not described herein again.

[0164] Step S703: Perform clustering processing according to the text representations of the text segments, to obtain a text clustering result.

[0165] After the text representations of the plurality of to-be-clustered text segments are obtained, a clustering algorithm is used, clustering is performed on the plurality of text segments according to a similarity between the text representations of the text segments, and text segments with similar semantics are clustered into a same type, to obtain the text clustering result.

[0166] The used clustering algorithm may be selected and configured according to a requirement in an actual application scenario. This is not specifically limited herein.

[0167] In this embodiment, the text representation model obtained in the foregoing embodiments is used in a text clustering task, high-quality text representations of the text segments may be obtained through the text representation model, and clustering is performed on the text segments based on the high-quality text representations, which can improve accuracy of text clustering.

[0168] FIG. 8 is a flowchart of a text retrieval method according to an exemplary embodiment of the present disclosure. This embodiment may be performed by an end-side device or a server with a text representation model running therein. As shown in FIG. 8, the method specifically includes the following steps.

[0169] Step S801: Obtain a query text in response to a query request.

[0170] In this embodiment, the query request is a request sent by a user through a user device, and the request includes the query text input by the user. After the query request is received, the query request is parsed to obtain the query text.

[0171] Step S802: Input the query text into a text representation model for encoding, to generate a text representation of the query text, where the text representation model is obtained through performing level classification on text data used for training to obtain a plurality of training sets of different data levels, and performing multi-stage training on a pre-trained model

by using a configured multi-stage training policy and the plurality of training sets of different data levels.

[0172] After the query text input by the user is obtained, the query text is input into the text representation model, and the query text is encoded into the text representation through the text representation model, that is, the text representation of the query text can be obtained.

[0173] The text representation model used in this embodiment is obtained through the method of the foregoing method embodiments: performing level classification on text data used for training, to obtain a plurality of training sets of different data levels; and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels. The text representation model has a good text representation capability and a good characteristic, which can improve quality of text representation. For a specific implementation and technical effect of obtaining the text representation model, refer to related content in the foregoing embodiments, and details are not described herein again.

[0174] Step S803: Retrieve, according to the text representation of the query text and text representations of candidate texts in a retrieval library, a candidate text that semantically matches the query text in the retrieval library, where the text representation of the candidate text is generated through encoding through the text representation model.

[0175] In this embodiment, the text representation model may be used to pre-construct the retrieval library. The retrieval library includes the candidate texts and the text representations of the candidate texts. When the retrieval library is constructed, the candidate texts are separately input into the text representation model for encoding, to obtain the text representations of the candidate texts.

[0176] When text retrieval is performed, according to the text representation of the query text and a text representation of each candidate text in the retrieval library, a similarity between the text representation of the query text and the text representation of each candidate text in the retrieval library is calculated as a semantic matching degree between the query text and the candidate text. At least one candidate text that has a relatively high semantic matching degree with the query text is retrieved from the retrieval library as the candidate text that semantically matches the query text.

[0177] For retrieving, based on the text representation, the candidate text that semantically matches the query text, any existing retrieval matching policy may be used, to determine the at least one candidate text that has a relatively high semantic matching degree with the query text. A specific retrieval matching policy may be configured according to an actual requirement of

a retrieval system. This is not specifically limited herein.

[0178] Step S804: Output the candidate text that semantically matches the query text.

[0179] After the candidate text that semantically matches the query text is retrieved, a retrieval result is output, to show the retrieval result to the user.

[0180] In this embodiment, the text representation model obtained in the foregoing embodiments is used in a text retrieval task, high-quality text representations of the query text and the candidate text may be obtained through the text representation model, and text retrieval is performed based on the high-quality text representations, which can improve accuracy of text retrieval.

[0181] Corresponding to the foregoing method, an embodiment of the present disclosure further provides a text data processing apparatus, including:

a classifying module, configured to perform level classification on text data used for training, to obtain a plurality of training sets of different data levels; and

a training module, configured to perform multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels, to obtain a text representation model, where the text representation model is used for generating a text representation of input text.

[0182] An embodiment of the present disclosure further provides a text representation apparatus, used in a server and including:

a receiving module, configured to receive an invoking request from an end-side device for a text representation model, where the invoking request includes to-be-processed text information;

an encoding module, configured to input the text information into the text representation model for encoding, to obtain a text representation of the text information, where the text representation model is obtained through performing level classification on text data used for training to obtain a plurality of training sets of different data levels, and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels; and

an outputting module, configured to output the text representation of the text information to the end-side device.

[0183] An embodiment of the present disclosure further provides a text clustering apparatus, including:

an obtaining module, configured to obtain a plurality of to-be-clustered text segments;

an encoding module, configured to separately input the plurality of text segments into a text representation model for encoding, to generate text representations of the text segments, where the text representation model is obtained through performing level classification on text data used for training to obtain a plurality of training sets of different data levels, and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels; and

a clustering module, configured to perform clustering processing according to the text representations of the text segments, to obtain a text clustering result.

[0184] FIG. 9 is a schematic structural diagram of an electronic device according to an embodiment of the present disclosure. As shown in FIG. 9, the electronic device includes: a memory 901 and a processor 902. The memory 901 is configured to store computer execution instructions, and may be configured to store various types of other data to support an operation on the electronic device. The processor 902 is communicatively connected to the memory 901, and is configured to execute the computer execution instructions stored in the memory 901, to implement the technical solution provided in any one of the foregoing method embodiments. A specific function is similar to an achievable technical effect, and details are not described herein. The electronic device may be a cloud server deployed on cloud, or may be a local server or terminal. That the electronic device is a cloud server is used as an example in FIG. 9. This is not specifically limited herein in this embodiment.

[0185] Optionally, as shown in FIG. 9, the electronic device further includes: a firewall 903, a load balancer 904, a communication component 905, a power supply component 906, and other components. Only some components are schematically shown in FIG. 9, which does not mean that the electronic device includes only the components shown in FIG. 9.

[0186] An embodiment of the present disclosure further provides a computer-readable storage medium. The computer-readable storage medium stores computer execution instructions, and the computer execution instructions, when executed by a processor, are used for implementing the technical solution provided in any one of the foregoing method embodiments. Details of a specific function and an achievable technical effect are not described herein.

[0187] An embodiment of the present disclosure further provides a computer program product. The computer program product includes a computer program. The computer program is stored in a readable storage medium, at least one processor of an electronic device may read the computer program from the readable storage medium, and the at least one processor executes the computer program to cause the electronic device to execute the technical solution

provided in any one of the foregoing method embodiments. Details of a specific function and an achievable technical effect are not described herein.

[0188] An embodiment of the present disclosure provides a chip, including: a processing module and a communication interface. The processing module can execute the technical solution of the electronic device in the foregoing method embodiments. Optionally, the chip further includes a storage module (such as a memory). The storage module is configured to store instructions, and the processing module is configured to execute the instructions stored in the storage module. In addition, the execution of the instructions stored in the storage module causes the processing module to execute the technical solution provided in any one of the foregoing method embodiments.

[0189] The foregoing memory may be object storage (Object storage service, OSS).

[0190] The memory may be implemented by any type of volatile or non-volatile storage devices or a combination thereof, such as a static random access memory (SRAM), an electrically erasable programmable read-only memory (EEPROM), am erasable programmable read-only memory (EPROM), a programmable read-only memory (PROM), a read-only memory (ROM), a magnetic memory, a flash memory, a magnetic disc, or an optical disc.

[0191] The communication component is configured to facilitate communication between a device on which the communication component is located and another device in a wired or wireless manner. The device on which the communication component is located may access to a wireless network based on a communication standard, such as a mobile communication network such as a mobile hotspot (WiFi), a second generation mobile communication system (2G), a third generation mobile communication system (3G), a fourth generation communication system (4G)/long term evolution (LTE), a fifth generation mobile communication system (5G), or a combination thereof. In an exemplary embodiment, the communication component receives a broadcast signal or broadcast-related information from an external broadcast management system through a broadcast channel. In an exemplary embodiment, the communication component further includes a near field communication (NFC) module, to promote short-range communication. For example, the NFC module may be implemented based on a radio frequency identification (RFID) technology, an infra-red data association (IrDA) technology, an ultra wideband (UWB) technology, a Bluetooth (BT) technology, or another technology.

[0192] The power supply component provides power for components of the device on which the power supply component is located. The power supply component may include a power supply management system, one or more power supplies, and other components related

to generation, management, and allocation of power for the device on which the power supply component is located.

[0193] A person skilled in the art should understand that the embodiments of the present disclosure may be provided as a method, a system, or a computer program product. Therefore, the present disclosure may use a form of hardware-only embodiments, software-only embodiments, or embodiments with a combination of software and hardware. Moreover, this specification may use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a disk memory, a compact disc read-only memory (CD-ROM), an optical memory, and the like) that include computer-usable program code.

[0194] The present disclosure is described with reference to the flowcharts and/or block diagrams of the method, the device (system), and the computer program product according to the embodiments of the present disclosure. It may be understood that, computer program instructions can implement each procedure and/or block in the flowcharts and/or block diagrams, and a combination of procedures and/or blocks in the flowcharts and/or block diagrams. These computer program instructions may be provided for a general-purpose computer, a dedicated computer, an embedded processor, or a processor of any other programmable data processing device to generate a machine, so that the instructions executed by a computer or a processor of any other programmable data processing device generate an apparatus for implementing a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0195] These computer program instructions may also be stored in a computer readable memory that can instruct the computer or any other programmable data processing device to work in a specific manner, so that the instructions stored in the computer-readable memory generate an artifact that includes an instruction apparatus. The instruction apparatus implements a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0196] These computer program instructions may also be loaded onto a computer or another programmable data processing device, so that a series of operations and steps are performed on the computer or the another programmable device, thereby generating computer-implemented processing. Therefore, the instructions executed on the computer or the another programmable device provide steps for implementing a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0197] In a typical configuration, a computing device includes one or more processors

(CPUs), an input/output interface, a network interface, and an internal memory.

[0198] The memory may include a form such as a volatile memory, a random-access memory (RAM) and/or a non-volatile memory such as a read-only memory (ROM) or a flash RAM in a computer-readable medium. The memory is an example of the computer-readable medium.

[0199] The computer-readable medium includes a non-volatile medium and a volatile medium, a movable medium and a non-movable medium, which may implement storage of information by using any method or technology. The information may be a computer-readable instruction, a data structure, a program module, or other data. Examples of computer storage media include, but are not limited to, a phase change memory (PRAM), a static random access memory (SRAM), a dynamic random access memory (DRAM), another type of random access memory (RAM), a read-only memory (ROM), an electrically erasable programmable read-only memory (EEPROM), a flash memory or another memory technology, a compact disc read-only memory (CD-ROM), a digital versatile disc (DVD) or another optical storage, a cassette magnetic tape, tape and disk storage or another magnetic storage device or any other non-transmission media that may be configured to store information that a computing device can access. Based on the definition in the present disclosure, the computer-readable medium does not include transitory computer-readable media (transitory media), such as a modulated data signal and a carrier.

[0200] It should be noted that, user information (including but not limited to user device information, user attribute information, and the like) and data (including but not limited to data used for analysis, stored data, displayed data, and the like) involved in the present disclosure are authorized by the user or fully authorized by all parties, and the collection, use, and processing of relevant data need to comply with relevant laws, regulations, and standards. In addition, a corresponding operation entrance is provided for a user to choose to authorize or reject.

[0201] In addition, some procedures described in the foregoing embodiments and accompanying drawings include a plurality of operations performed in a particular order. However, it should be clearly learned that these operations may not be performed in the order in which these operations are performed in the specification or concurrently performed. Sequence numbers are merely used to distinguish different operations, and do not represent any execution order. In addition, these procedures may include more or fewer operations, and these operations may be performed in an order or may be performed concurrently. It should be noted that, descriptions of "first", "second", and the like in the specification are used to distinguish

different messages, devices, modules, and the like, and do not indicate a sequence and do not limit that "first" and "second" are different types either. "A plurality of" means two or more, unless otherwise definitely and specifically limited.

[0202] Other implementation solutions of the present disclosure will be apparent to a person skilled in the art from consideration of the specification and practice of the content disclosed here. The present disclosure is intended to cover any variation, use, or adaptive change of the present disclosure. These variations, uses, or adaptive changes follow the general principles of the present disclosure and include common general knowledge or common technical means in the art that are not disclosed in the present disclosure. The specification and the embodiments are considered as exemplary only, and a true scope and spirit of the present disclosure are indicated by the following claims.

[0203] The present disclosure is not limited to the precise structures described above and shown in the accompanying drawings, and various modifications and changes can be made without departing from the scope of the present disclosure. The scope of the present disclosure is subject only to the appended claims.

CLAIMS

1. A text data processing method, comprising:

performing level classification on text data used for training, to obtain a plurality of training sets of different data levels; and

performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels, to obtain a text representation model, wherein the text representation model is used for generating a text representation of input text.

2. The method according to claim 1, wherein the performing level classification on text data used for training, to obtain a plurality of training sets of different data levels comprises:

calculating use indicators of the text data according to use behavior data of source data of the text data; and

classifying the text data into the plurality of training sets of different data levels according to the use indicators of the text data, wherein a use indicator of text data in a training set of a high data level is greater than a use indicator of text data in a training set of a low data level.

3. The method according to claim 1 or 2, wherein the text data comprises a plurality of text pairs, and the text pair comprises two content-related texts, and the performing level classification on text data used for training, to obtain a plurality of training sets of different data levels comprises:

classifying the text data into the plurality of training sets of different data levels according to correlations of the text pairs, wherein a correlation of a text pair in a training set of a high data level is greater than a correlation of a text pair in a training set of a low data level.

4. The method according to claim 3, further comprising:

collecting a plurality of text corpora of different data types; and

extracting, according to the data types of the text corpora, the text pairs from the text corpora as the text data used for training.

5. The method according to any one of claims 1 to 4, wherein the performing level classification on text data used for training, to obtain a plurality of training sets of different data levels comprises:

according to whether the text data is annotated, classifying unannotated text data into a weakly supervised training set, and classifying annotated text data into a strongly supervised training set, wherein a data level of the weakly supervised training set is lower than a data level of the strongly supervised training set.

6. The method according to any one of claims 1 to 5, wherein the configured multi-stage training policy comprises a configured quantity of training stages and training sets used in the training stages; and the performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels, to obtain a text representation model comprises:

training the pre-trained model by using the configured multi-stage training policy and sequentially using the training sets corresponding to the training stages, to obtain the text representation model.

7. The method according to any one of claims 1 to 6, wherein the performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels, to obtain a text representation model comprises:

performing, according to the plurality of training sets of different data levels, stage training on the pre-trained model by sequentially using the training sets in ascending order of data level, to obtain the text representation model.

8. The method according to claim 6 or 7, wherein the text data comprises a plurality of text pairs, and the text pair comprises two content-related texts; and the method further comprising:

when performing any stage of training on the pre-trained model, based on a text pair in a training set used in a current stage, for a first text in any text pair, using a second text belonging to a same text pair as the first text as a positive sample relevant to the first text, using a second text belonging to a different text pair from the first text as a negative sample irrelevant to the first text, and performing contrastive learning training on the pre-trained model.

9. The method according to claim 6 or 7, further comprising:

outputting, in response to a configuration/update request of the configured multi-stage training policy, the configured multi-stage training policy, data levels of the training sets, and description information of the training sets through a training policy configuration interface; and

configuring/updating the configured multi-stage training policy in response to an editing operation on any configuration information in the configured multi-stage training policy, wherein the configuration information comprises: a quantity of training stages, and training sets used in the training stages;

wherein the method further comprises:

outputting a current training result through a front-end interface after performing any stage of training on the pre-trained model, wherein the current training result is used for guiding

adjustment of the configured multi-stage training policy; and

using, in response to an update of the configured multi-stage training policy, an updated multi-stage training policy to perform a subsequent stage of training on the pre-trained model.

10. The method according to any one of claims 1 to 7, wherein after the performing multi-stage training on a pre-trained model by using the plurality of training sets of different data levels, to obtain a text representation model, the method further comprises:

receiving a dedicated training set of a vertical field submitted by an end-side device;

using the dedicated training set to train the text representation model, to obtain a text representation model suitable for the vertical field; and

providing the text representation model suitable for the vertical field for the end-side device.

11. A text representation method, applied to a server and comprising:

receiving an invoking request from an end-side device for a text representation model, wherein the invoking request comprises to-be-processed text information;

inputting the text information into the text representation model for encoding, to obtain a text representation of the text information, wherein the text representation model is obtained through performing level classification on text data used for training to obtain a plurality of training sets of different data levels, and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels; and

outputting the text representation of the text information to the end-side device.

12. A text clustering method, comprising:

obtaining a plurality of to-be-clustered text segments;

separately inputting the plurality of text segments into a text representation model for encoding, to generate text representations of the text segments, wherein the text representation model is obtained through performing level classification on text data used for training to obtain a plurality of training sets of different data levels, and performing multi-stage training on a pre-trained model by using a configured multi-stage training policy and the plurality of training sets of different data levels; and

performing clustering processing according to the text representations of the text segments, to obtain a text clustering result.

13. An electronic device, comprising: a processor and a memory communicatively connected to the processor, wherein

the memory stores computer execution instructions; and

the processor executes the computer execution instructions stored in the memory, to implement the method according to any one of claims 1 to 12.

14. A computer-readable storage medium, storing computer execution instructions, wherein when executed by a processor, the computer execution instructions are used for implementing the method according to any one of claims 1 to 12.

15. A computer program product, comprising a computer program, wherein when the computer program is executed by a computer, the method according to any one of claims 1 to 12 is implemented.