Text Processing Method, Electronic Device, and Storage Medium

TECHNICAL FIELD

[0001] The present application relates to the field of large model technology and data processing. Specifically, it pertains to a text processing method, electronic device, and storage medium.

BACKGROUND

[0002] Large Language Models (LLM) can be applied to various Natural Language Processing (NLP) tasks, thereby helping people to process natural language texts more quickly and accurately. Although LLMs perform well in various NLP tasks, most of these models primarily handle high-resource language texts, such as English or Chinese. When it comes to processing low-resource and regional languages (e.g., Southeast Asian languages), which have scarce data and high tokenization difficulty, these models tend to exhibit lower tokenization efficiency and poor performance. This, in turn, affects the model's text understanding and generation capabilities.

[0003] Currently, there have been no effective solutions proposed to address the aforementioned issues.

SUMMARY

[0004] This embodiments of the present application provide a text processing method, an electronic device, and a storage medium, aiming to at least solve the technical problem of poor text understanding and generation performance for low-resource languages in related technologies.

[0005] According to one aspect of the embodiments of the present application, a text processing method is provided, including: obtaining text to be processed, wherein text content of the text to be processed is described in a first language, and the first language is a

regional language; using a first language model to perform text understanding and generation on the text to be processed to obtain a text processing result, wherein the first language model is a large language model obtained by performing language enhancement on the first language based on a second language model, the second language model is a large language model applicable to a second language, and the second language is a universal language.

[0006] According to another aspect of the embodiments of the present application, a text processing method is also provided, including: performing vocabulary expansion on an initial tokenization method to obtain a target tokenization method; training a universal language model based on the target tokenization method to obtain a regional language model; using the regional language model to perform text understanding and generation on text to be processed to obtain a text processing result; wherein the text content of the text to be processed is described in a regional language, and the regional language model is a large language model obtained by performing language enhancement on the regional language based on the universal language model, and the universal language model is a large language model suitable for a universal language.

[0007] According to another aspect of the embodiments of the present application, an electronic device is also provided, including: a memory storing executable programs; and a processor to run the programs, wherein the programs, when executed, perform the text processing method in the embodiments of this application.

[0008] According to another aspect of the embodiments of the present application, a computer-readable storage medium is also provided. The computer-readable storage medium includes stored executable programs, which, when executed, control a device where the computer-readable storage medium is located to perform the text processing method of any one of the embodiments of this application.

[0009] In the embodiments of this application, by obtaining the text to be processed, where the text content of the text to be processed is described in a first language, the first language being a regional language, and then using a first language model to perform text understanding and generation on the text to be processed to obtain a text processing result, wherein the first language model is a large language model obtained by performing language enhancement on the first language based on a second language model, and the second

language model is a large language model suitable for a second language, the second language being a universal language, the objective of accurately understanding and generating regional languages is achieved. This realizes the technical effect of improving the understanding and generation of regional languages, thereby solving the technical problem of poor text understanding and generation for low-resource languages in related technologies.

[0010] It is noteworthy that the above general description and the following detailed description are merely provided for illustration and explanation of this application and do not constitute a limitation on this application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The accompanying drawings, which are included to provide a further understanding of this application, are incorporated in and constitute a part of this application. The illustrative embodiments and their descriptions are intended to explain, but not unduly limit, the scope of this application. In the drawings:

[0012] FIG. 1 is a schematic diagram of an application scenario of a text processing method according to Embodiment 1 of this application;

[0013] FIG. 2 is a flowchart of a text processing method according to embodiment 1 of this application;

[0014] FIG. 3 is a flowchart of training a first language model according to embodiment 1 of this application;

[0015] FIG. 4 is a flowchart of a text processing method according to embodiment 2 of this application;

[0016] FIG. 5 is a structural block diagram of a text processing device according to embodiment 1 of this application;

[0017] FIG. 6 is a structural block diagram of a text processing device according to embodiment 2 of this application;

[0018] FIG. 7 is a structural block diagram of a computer terminal according to embodiment 1 of this application.

DETAIL DESCRIPTION OF THE EMBODIMENTS

[0019] To enable those skilled in the art to better understand the solutions of this application, the technical solutions in the embodiments of this application will be clearly and comprehensively described below in conjunction with the accompanying drawings in the embodiments of this application. It is evident that the described embodiments are merely part of the embodiments of this application and not all of them. Based on the embodiments in this application, all other embodiments obtained by those of ordinary skill in the art without making creative efforts shall fall within the scope of protection of this application.

[0020] It should be noted that the terms "first," "second," and the like in the description and claims of this application and the above drawings are used to distinguish similar objects and are not necessarily used to describe a particular order or sequence. It should be understood that the data so used are interchangeable under appropriate circumstances so that the embodiments of the application described herein can be practiced in sequences other than those illustrated or described herein. Furthermore, the terms "comprising" and "having," and any variations thereof, are intended to cover non-exclusive inclusions, for example, a process, method, system, product, or apparatus that includes a series of steps or components is not necessarily limited to those steps or components explicitly listed but may include other steps or components that are not explicitly listed or inherent to such process, method, product, or apparatus.

[0021] The technical solutions provided by this application primarily utilize large model technology. Here, a large model refers to a deep learning model with a large number of model parameters, which can typically include billions, tens of billions, hundreds of billions, trillions, or even tens of trillions of model parameters. Large models are also known as foundation models. These models are pre-trained on large-scale unannotated corpora to produce pre-trained models with billions or more parameters. Such models can adapt to a wide range of downstream tasks and exhibit good generalization capabilities. Examples include large language models (LLMs) and multi-modal pre-training models.

[0022] It should be noted that in practical applications, large models can be fine-tuned with a small number of samples, enabling them to be applied to various tasks. For instance, large

models can be widely used in fields such as Natural Language Processing (NLP), computer vision, and speech processing. Specifically, they can be applied to tasks in the computer vision domain, such as Visual Question Answering (VQA), Image Captioning (IC), and image generation. They can also be broadly used in the NLP domain for tasks like sentiment classification based on text, text summarization, and machine translation. Therefore, the main application scenarios for large models include, but are not limited to, digital assistants, intelligent robots, search engines, online education, office software, e-commerce, and intelligent design. In the embodiments of this application, the explanation will be illustrated with an example of data processing using a large language model in the context of text processing.

[0023] Firstly, some terms or phrases that appear in the description of the embodiments of this application are to be interpreted as follows:

[0024] **Large Language Model (LLM):** a machine learning model used for natural language processing.

[0025] **Pre-train:** refers to training using unsupervised text.

[0026] **Supervised Fine-Tuning (SFT):** refers to the further adjustment of model weights using (instruction, response) type data, where the instruction can be on any topic and the response is the desired answer to the instruction. By using such data to train different types of models, the model can be guided to respond to user instructions in a specific form or according to particular requirements.

[0027] In related technologies, large language models are also used to handle various NLP tasks. However, because these large language models are primarily based on English, their performance on various types of languages (e.g., Southeast Asian languages) is suboptimal, and their tokenization efficiency for these languages is low. Other open-source or closed-source models based on existing large language models might have been fine-tuned for specific languages or tasks, but they still cannot effectively handle the aforementioned various types of languages (e.g., Southeast Asian languages). Therefore, from the perspective of language scope, there is currently no large language model specifically designed to consider these various types of languages (e.g., Southeast Asian languages).

[0028] In summary, it is evident that related technologies face the technical problem of poor

text understanding and generation for low-resource languages. As of now, no effective solution has been proposed to address this issue.

[0029] Embodiment 1

[0030] According to an embodiment of the present application, a text processing method is provided. It should be noted that the steps shown in the flowchart of the accompanying drawings can be executed in a computer system, such as a set of computer-executable instructions. Although the logical sequence is shown in the flowchart, the steps illustrated or described can, in some cases, be executed in an order different from that shown or described here.

[0031] Considering the large number of parameters in a large language model and the limited computational resources of mobile terminals, the text processing method provided in this embodiment can be applied to the application scenario shown in FIG. 1, but is not limited to this. In the application scenario shown in **FIG. 1**, the large language model is deployed on a server 10. The server 10 can be connected to one or more client devices 20 through a local area network (LAN), wide area network (WAN), internet connection, or other types of data networks. The client devices 20 may include, but are not limited to, smartphones, tablets, laptops, handheld computers, personal computers, smart home devices, in-car devices, and more. The client devices 20 can interact with users through a graphical user interface to call the large language model, thereby implementing the method provided by the embodiments of this application.

[0032] In the embodiments of this application, the system composed of the client device and the server can execute the following steps: the client device, in response to a first touch operation on the graphical user interface, inputs the text to be processed and selects a first language from a plurality of candidate languages, where the first language is a regional language, thereby enabling the server to obtain the text to be processed; the client device also responds to a second touch operation on the graphical user interface, prompting the server to use the first language model to perform text understanding and generation on the text to be processed, resulting in a text processing result, which is then returned to the client device. The client device can display the text processing result within the graphical user interface. It should be noted that if the operating resources of the client device meet the deployment and

operating requirements of the large language model, the embodiment of this application can be executed on the client device.

[0033] In the above operating environment, this application provides a text processing method as shown in FIG. 2. FIG. 2 is a flowchart of a text processing method according to Embodiment 1 of this application. As shown in FIG. 2, the method may include the following

[0034] **S21:** obtaining text to be processed, wherein the text content of the text to be processed is described in a first language, and the first language is a regional language.

[0035] **S22:** using a first language model to perform text understanding and generation on the text to be processed to obtain a text processing result. The first language model is a large language model obtained by performing language enhancement on the first language based on a second language model, the second language model is a large language model suitable for the second language, and the second language is a universal language.

[0036] The text content to be processed is described in the first language, which is a regional language. The regional language is a low-resource language with scarce data and high segmentation difficulty. A low-resource language refers to a language used by a relatively small number of people globally, typically spoken by smaller communities or regions, and may lack written form or standardized linguistic norms. Specifically, the regional language can be any type of language other than the internationally common languages such as English, Chinese, French, Russian, Spanish, and Arabic (also known as minor languages), for example, Southeast Asian languages.

[0037] By way of example, in the process of obtaining the text to be processed, the text data can be collected from various sources such as the internet, text databases, news content, academic publications, and professional knowledge bases. Subsequently, the collected text data undergoes cleaning, which includes removing special characters, punctuation marks, and stop words, as well as performing stemming and lemmatization operations. The processed text data is then stored in a database or file for access and processing in subsequent steps.

[0038] The first language model is a large language model obtained by performing language enhancement on the first language based on a second language model. The second language model is a large language model suitable for the second language, which is a universal language. A universal language is a high-resource language, referring to a language spoken

by a large number of people globally, often used as an official language or for international communication. Universal languages typically have a written form and standardized linguistic norms, and they possess richer linguistic resources, including texts, literary works, and educational materials. Examples include English, Chinese, French, Russian, Spanish, and Arabic.

[0039] It should be noted that in the embodiments of this application, the implementation process of the text processing method is illustrated using English as an example of the universal language. However, this does not constitute a specific limitation on the choice of the universal language.

[0040] The text understanding and generation refer to the use of natural language processing techniques to analyze and comprehend text, and then generate new text based on the understood content. This process can utilize the first language model. The first language model is a large language model obtained by performing language enhancement on the regional language based on a second language model. The second language model is a large language model suitable for universal languages, such as the English-based model llama2.

[0041] By way of example, during the text understanding process, the first language model can perform lexical analysis, syntactic analysis, and semantic analysis on the text to be processed, thereby understanding the meaning and structure of the text. Through this process, the first language model can identify entities, relationships, events, and other information within the text, and extract and summarize these elements. In the text generation process, the first language model generates new text that conforms to grammatical and semantic rules based on the understood content. This process can be applied in natural language generation, question-answering systems, and dialogue systems. The text understanding and generation process can help people quickly obtain text information, improve information processing efficiency, and can also be used in fields such as machine translation, intelligent customer service, and intelligent writing, providing intelligent text processing services.

[0042] By enhancing the first language based on the second language model, text understanding and generation of the text to be processed are achieved. By using the first language model and the second language model, the text to be processed can be understood and generated more accurately, thereby improving the accuracy and fluency of text processing.

Additionally, this method can facilitate text processing and translation between different languages, thereby increasing the efficiency and accuracy of regional language processing.

[0043] Based on the aforementioned S21 to S22, by obtaining the text to be processed, wherein the text content is described in the first language, which is a regional language, and subsequently using the first language model to perform text understanding and generation on the text to be processed, the text processing result is obtained. The first language model is a large language model obtained by performing language enhancement on the first language based on a second language model. The second language model is a large language model suitable for the second language, which is a universal language. This approach achieves the purpose of accurately understanding and generating regional languages, thereby improving the understanding and generation effect of regional languages. Consequently, it addresses the technical problem of poor text understanding and generation effects for minor languages in related technologies.

[0044] Optionally, the text processing method provided in this embodiment of the application can be applied, but is not limited to, fields such as natural language processing, text generation, intelligent customer service, finance, healthcare, and education. Specifically, it can be applied to scenarios involving the understanding and generation of text in regional languages.

[0045] The following provides a further introduction to the text processing method in the embodiments of this application.

[0046] In an alternative embodiment, FIG. 3 illustrates a training flowchart of a first language model according to Embodiment 1 of this application. As shown in FIG. 3, the text processing method in this embodiment further includes:

[0047] **S31:** performing vocabulary expansion on an initial tokenization method to obtain a target tokenization method;

[0048] **S32:** pre-training the second language model based on the target tokenization method to obtain a first training result;

[0049] **S33:** performing a mixture of pre-training and supervised fine-tuning on the first training result based on the target tokenization method to obtain a second training result;

[0050] **S34:** performing supervised fine-tuning on the second training result based on the

target tokenization method to obtain the first language model.

[0051] The initial tokenization method can be achieved using the original tokenizer, and the target tokenization method can be realized using the target tokenizer. By extending the vocabulary of the original tokenizer, the target tokenizer can be obtained. A tokenizer is a tool used to segment text or language data into words or phrases. It can divide a continuous text stream into meaningful words, typically used in natural language processing, information retrieval, and text analysis. Tokenizers can be customized according to different languages and requirements to suit various text processing tasks.

[0052] The above first training result is the output of the pre-training process. The second training result is the output of the pre-training and supervised fine-tuning hybrid process (Pre-train & SFT hybrid). The first language model is obtained by applying SFT to the second training result based on the target tokenization method.

[0053] Based on the above optional embodiments, the target tokenization method is obtained by extending the vocabulary of the initial tokenization method. Subsequently, the first training result is obtained by pre-training the second language model based on the target tokenization method. Then, the second training result is obtained by applying a hybrid of pre-training and supervised fine-tuning to the first training result based on the target tokenization method. Finally, the first language model is achieved by applying supervised fine-tuning to the second training result based on the target tokenization method. This process enables the first language model to better adapt to the characteristics of regional languages, enhancing the model's generalization and language understanding capabilities. The performance and accuracy of the first language model in text processing tasks can be improved through supervised fine-tuning.

[0054] In an optional embodiment, in S31, extending the vocabulary of the initial tokenization method to obtain the target tokenization method includes:

[0055] **S311:** selecting a document to be tokenized from a document sequence, where the document to be tokenized is described in a first language;

[0056] **S312:** tokenizing the document to be tokenized according to the initial tokenization method to obtain a first tokenization result;

[0057] **S313:** determining target extended vocabulary based on the first tokenization result, a first word set, and the second word set, where the first word set comprises a plurality of first

language words, and the second word set comprises a plurality of second language words;

[0058] S314: determining a target word set using the second word set and the target extended vocabulary;

[0059] S315: determining the target tokenization method based on the target word set.

[0060] Specifically, selecting a document d from the document sequence D, where the document is described in multiple types of languages (e.g., Southeast Asian languages). Tokenizing the document to be tokenized according to the initial tokenization method to obtain the first tokenization result t_V. Then, determining the target extended vocabulary T based on the first tokenization result t_V, the first word set Vt that includes a plurality of words from various types of languages (e.g., Southeast Asian languages), and the second word set Vi that includes a plurality of English words. Using the second word set Vi and the target extended vocabulary T to determine the target word set V_{final}. Finally, determining the target tokenization method through the target word set V_{final}.

[0061] Based on the above optional embodiment, by selecting a document to be tokenized from the document sequence and then tokenizing it according to the initial tokenization method to obtain the first tokenization result, the process can determine the target extended vocabulary based on the first tokenization result, the first word set, and the second word set. Using the second word set and the target extended vocabulary, the target word set is then determined. Finally, the target tokenization method is established through the target word set. This approach better meets the semantic requirements of the documents, improving the efficiency and accuracy of text processing.

[0062] In an optional embodiment, in step 313, determining the target extended vocabulary based on the first tokenization result, the first word set, and the second word set includes:

[0063] S3131: determining candidate extended vocabulary based on the first tokenization result, the first word set, and the second word set;

[0064] S3132: selecting extended vocabulary that meets preset conditions from the candidate extended vocabulary to obtain the target extended vocabulary, where the preset conditions are determined by the occurrence frequency of each word in the candidate extended vocabulary.

[0065] Specifically, determining the candidate extended vocabulary t_V based on the first

tokenization result V_t, the first word set $V_{t,}$ and the second word set V_i. Then, selecting the extended vocabulary from T_{new} that meets the condition where the occurrence frequency f_t of the vocabulary is greater than or equal to the preset threshold m, thereby obtaining the target extended vocabulary.

[0066] Based on the above optional embodiment, by determining the candidate extended vocabulary based on the first tokenization result, the first word set, and the second word set, and then selecting the extended vocabulary that meets the preset conditions from the candidate extended vocabulary to obtain the target extended vocabulary, this method can quickly and accurately identify extended vocabulary related to the original words. This improves the efficiency and accuracy of information retrieval and language processing. Additionally, it can help users discover new vocabulary and expressions, enriching language resources and enhancing the expressiveness of the text.

[0067] In an optional embodiment, in S3131, determining the candidate extended vocabulary based on the first tokenization result, the first word set, and the second word set includes:

[0068] **S31311:** selecting a token pair from the first tokenization result, where the token pair comprises two adjacent words;

[0069] **S31312:** merging the token pair to obtain a merged result;

[0070] **S31313:** if the merged result exists in the first word set, replacing the token pair in the first tokenization result with the merged result, and adding the merged result to the candidate extended vocabulary.

[0071] Specifically, selecting token pairs (prev, next) from the first tokenization result tv, and merging the token pairs (prev, next) to obtain the merged result t_{merged}. If the merged result t_{merged} exists in the first word set V_t, replacing the token pair (prev, next) in the first tokenization result t_V with the merged result t_{merged}, and adding the merged result t_{merged} to the candidate extended vocabulary.

[0072] Based on the above optional embodiment, by selecting token pairs from the first tokenization result and then merging these token pairs to obtain a merged result, the process can improve the accuracy and semantic coherence of tokenization. If the merged result exists in the first word set, it replaces the token pair in the first tokenization result and is added to

the candidate extended vocabulary. This method enhances the precision and semantic consistency of tokenization.

[0073] In an optional embodiment, in S32, pre-training the second language model based on the target tokenization method to obtain the first training result includes:

[0074] **S321:** obtaining a pre-training dataset based on the first language and data quality, where the pre-training dataset comprises a plurality of first documents and a plurality of second documents, the data quality of the plurality of first documents is higher than that of the plurality of second documents, both the plurality of first documents and the plurality of second documents are selected from data streams corresponding to the first language, with different languages corresponding to independent data streams;

[0075] **S322:** tokenizing the pre-training dataset according to the target tokenization method to obtain the second tokenization result;

[0076] **S323:** pre-training the second language model using the second tokenization result to obtain the first training result.

[0077] Specifically, acquiring a pre-training dataset based on the first language and data quality, where the pre-training dataset includes a plurality of first documents and a plurality of second documents. The first documents contain high-quality data, and the second documents contain low-quality data. High-quality data refers to data that is accurate, complete, consistent, reliable, and useful, which helps the model make accurate decisions and supports effective analysis. Low-quality data, on the other hand, is inaccurate, incomplete, inconsistent, unreliable, or useless, potentially leading to erroneous decisions and analysis results. The plurality of first documents and a plurality of second documents are selected from data streams corresponding to various types of languages (e.g., Southeast Asian languages), with different languages each corresponding to independent data streams L_i.

[0078] Furthermore, tokenizing the pre-training dataset according to the target tokenization method to obtain the second tokenization result. Subsequently, using the second tokenization result to pre-train the Llama2 model, resulting in the first training outcome.

[0079] Based on the above optional embodiment, by acquiring a pre-training dataset based on the first language and data quality, and then tokenizing the pre-training dataset according to the target tokenization method to obtain the second tokenization result, the second language

model is pre-trained using the second tokenization result to obtain the first training result. This process enhances the applicability and performance of the second language model in regional languages.

[0080] The pre-training process is explained in conjunction with a specific embodiment as follows.

[0081] Pre-training data includes various types of publicly available documents, such as corpora collected from the internet. During the first stage of the pre-training process, some commonly used international languages are selected. For example, using a language recognition tool (Fasttext), documents in English, Chinese, Thai, Vietnamese, or Indonesian are chosen, and all other languages are discarded. In the second stage of the pre-training process, on the basis of the languages already selected in the first stage, some less commonly used languages are added. For instance, pre-training data for Khmer, Lao, Malay, Burmese, and Tagalog is incorporated. The aforementioned language recognition tool (Fasttext) is an efficient text classification tool that utilizes the Fasttext model. This model is a word vector-based text classification model that represents words in the text using character-level features, thereby avoiding the need for the bag-of-words model to consider all possible word sequences.

[0082] To enhance the model's tokenization efficiency for non-European and non-Latin scripts in the Byte Pair Encoding (BPE) tokenizer, the initial tokenization method is expanded to obtain a target tokenization method. This introduces a minimal number of additional language elements (tokens) for newly added languages. Additionally, through vocabulary expansion, the encoding compression ratio for various types of languages (e.g., Southeast Asian languages) can be adjusted to be nearly equivalent to that of English. By retaining the token structure of Latin character languages, the existing knowledge in the lama-2 model is preserved.

[0083] Furthermore, the pre-training dataset is organized based on the language of the text and the quality of the data. For each language, an independent data stream LiL_iLi can be established in advance. When pre-training documents are needed, a predefined probability is used to select one language's data stream, and then a plurality of first documents and a plurality of second documents are selected from that data stream. It is important to note that the document selection process occurs at each step of updating the model parameters. This

approach allows the aggregation of documents in different languages into a single training sequence, ensuring an efficient training process.

[0084] In the first stage of the pre-training process, each language-specific data stream provides specialized high-quality documents for the second language model, thereby helping to guide the learning process towards a preference for high-quality information. If all high-quality data has been used, the data stream can also provide a mixed dataset of high-quality and low-quality documents. In the second stage of the pre-training process, the model undergoes additional thousands of steps of pre-training, reverting to the use of only high-quality documents. This readjusts the model's learning focus back to high-quality data, thus enhancing the model's performance.

[0085] In an alternative embodiment, in S33, a mixture of pre-training and supervised fine-tuning based on the target tokenization method is applied to the first training results to obtain the second training results, which include:

[0086] **S331:** mixing the pre-training dataset, annotated data for natural language processing tasks, and instruction-following data to obtain a mixed dataset;

[0087] **S332:** tokenizing the mixed dataset according to the target tokenization method to obtain the third tokenization result;

[0088] **S333:** performing a mixture of pre-training and supervised fine-tuning on the first training result using the third tokenization result to obtain the second training result.

[0089] Specifically, when developing a language model that responds to instructions, the related technology can fine-tune a pre-trained model using a dataset that contains instructions or conversation content. However, in this application, a pre-training-SFT (supervised fine-tuning) mixed step is added before fine-tuning. In this step, the model is further trained using a mixed dataset that includes three types of data: the original pre-training corpus, annotated data from traditional natural language processing tasks, and instruction-following data. Compared to the instruction data for a plurality of language types (e.g., Southeast Asian languages), the quantity of open-source English instruction-following data is relatively large. During the supervised fine-tuning process, this abundance of data may overshadow the smaller datasets of instruction data for a plurality of language types (e.g., Southeast Asian languages). Therefore, it is necessary to mix various datasets to create a more diverse mixed

dataset.

[0090] In the mixed training phase, different types of data can be combined into batches of a single sequence to form a mixed dataset. A single sequence may consist of general pre-training content and instruction-following examples. This mixed processing differs from the initial pre-training phase, where the primary goal of mixed processing was to avoid truncating unsupervised text in a way that affects its integrity. Using a mixed dataset helps prevent the model from overfitting to training examples and reduces the risk of the model simply memorizing input data rather than learning a more general capability to follow instructions.

[0091] Based on the above alternative embodiment, by mixing the pre-training dataset, annotated data for natural language processing tasks, and instruction-following data, a mixed dataset is obtained. Then, the mixed dataset is tokenized according to the target tokenization method to obtain the third tokenization results. Finally, the third tokenization results are used to perform a mixture of pre-training and supervised fine-tuning on the first training results, yielding the second training results. This approach enhances the generalization ability and adaptability of the language model.

[0092] In an alternative embodiment, in S34, supervised fine-tuning is performed on the second training results based on the target tokenization method to obtain the first language model, which includes:

[0093] **S341:** obtaining supervised data of various categories, where the supervised data includes at least a part or all of the following data: text task processing data, logical reasoning data, instruction data generated through self-instruction, natural conversation data, and safety data;

[0094] **S342:** tokenizing the supervised data according to the target tokenization method to obtain the fourth tokenization result;

[0095] **S343:** performing supervised fine-tuning on the second training result using the fourth tokenization result to obtain the first language model.

[0096] Specifically, the supervised data includes multiple categories to enable the model's instruction-following capability. The supervised data comprises at least some or all of the following: text task processing data, such as text classification, summary generation, etc.;

logical reasoning data, such as mathematical problems and logical reasoning questions; Instruction data generated through self-instruction, such as general instructions and responses obtained via the self-instruct method; Natural conversation data: considering the extensive background knowledge behind various types of languages (e.g., Southeast Asian languages), natural conversation data is collected from local forums. The original questions from users result in dialogue data that is more aligned with local characteristics and expressive styles; safety data: for each region-specific topic, corresponding safety input-output pairs are constructed to ensure the model can respond appropriately to safety-related topics specific to different countries.

[0097] Based on the above optional embodiments, various categories of supervised data are obtained, which are then tokenized according to the target tokenization method to obtain the fourth tokenization result. Subsequently, the fourth tokenization result is used to perform supervised fine-tuning on the second training result, yielding the first language model. This process enhances the generalization ability and adaptability of the language model.

[0098] In an optional embodiment, in S34, performing supervised fine-tuning on the second training result based on the target tokenization method to obtain the first language model includes:

[0099] **S344**: generating a fine-tuning dataset based on a first prompt template, where the fine-tuning dataset comprises a plurality of first data pairs, each of the plurality of first data pairs includes a matching question and answer, and the plurality of first data pairs are constructed from the questions and fact-based documents associated with the questions;

[0100] **S345**: tokenizing the fine-tuning dataset according to the target tokenization method to obtain a fourth tokenization result;

[0101] **S346**: performing supervised fine-tuning on the second training result using the fourth tokenization result to obtain the first language model.

[0102] Specifically, the fine-tuning dataset is generated by controlling the pre-trained language model based on the first prompt template. This fine-tuning dataset can be used as SFT (Supervised Fine-Tuning) samples added to the final SFT data pool. The fine-tuning dataset includes a plurality of first data pairs, each being a Question and Answer (QA) pair, meaning each first data pair consists of a matching question and answer. These data pairs are

constructed from questions and their associated factual documents, which may be high-quality documents collected from various information sources. Furthermore, the fine-tuning dataset is tokenized according to the target tokenization method to obtain the fourth tokenization result. The fourth tokenization result is then used to perform supervised fine-tuning on the second training result, yielding the first language model.

[0103] Based on the knowledge learned by the model during pre-training, the model can enhance the SFT (Supervised Fine-Tuning) data by comprehensively generating QA pairs during the context-based QA data generation process. High-quality document fragments are collected from various internet knowledge query pages, and the initial SFT model generates a question and a correct answer based on the knowledge cues provided in the document. Different prompt templates can be used to generate different types of questions, such as standard knowledge queries, multiple-choice questions, or advanced critical thinking questions.

[0104] After collecting a large amount of context-based QA data, the context documents are removed, and the QA pairs are solely considered as SFT (Supervised Fine-Tuning) samples to be added to the final SFT data pool. Removing the context documents helps to force the model to recall the internet-sourced knowledge it has seen during pre-training and improve its own knowledge base without providing the contextual knowledge.

[0105] Based on the above optional embodiments, the fine-tuning dataset is generated using the first prompt template. The fine-tuning dataset is then tokenized according to the target tokenization method to obtain the fourth tokenization result. Finally, the fourth tokenization result is used to perform supervised fine-tuning on the second training result, yielding the first language model. This process enhances the effectiveness and accuracy of natural language processing.

[0106] In an optional embodiment, the text processing method in this application further includes:

[0107] **S35:** generating a preference optimization dataset by controlling the first language model using the second prompt template. The preference optimization dataset comprises a plurality of second data pairs, each of the plurality of second data pairs includes a matching input instruction and a target response, and the target response is selected from a plurality of

candidate responses corresponding to the input instructions;

[0108] **S36:** performing preference adjustment on the first language model using the preference optimization dataset to obtain the adjusted first language model.

[0109] Specifically, the second data pairs are instruction-response pairs. To further enhance the model's practicality and safety, pre-set system prompts are integrated into each instruction sample. These system prompts are comprehensive and adhere to safety considerations. By incorporating detailed system prompts, a useful and safe prior can be established within the model, contributing to overall performance improvement and increasing adherence to safety protocols. In the later stages of fine-tuning, multiple rounds of dialogue are artificially created by randomly linking several single-turn instructions, thereby significantly improving the model's conversational capabilities through synthetic tuning.

[0110] Using the current model, generate a plurality of different candidate responses for the same input instruction. These candidate responses may be similar to each other or may vary in quality. The target response is then selected from these plurality of candidate responses corresponding to the input instruction.

[0111] For example, given an input instruction, the current model can be prompted to evaluate its own response. This can be done using the following second prompt template: "You are an expert human judge whose job is to rate and compare the responses of two different AI assistants to the same prompt. Below is the prompt and the two responses from Assistant 1 and Assistant 2. You must indicate whether you prefer Assistant 1 or Assistant 2. If both are equally good, mark it as a tie. If both are poor, mark neither as preferred. Here are some guidelines for judgment criteria: a preferred response should be safe, true, accurate, and satisfactorily answer the prompt while maintaining safety, and should be coherent, comprehensive, and well-structured. A poor response may be unsafe, inaccurate, incorrect, contain false information, fail to accurately address the prompt, be too short or irrelevant, incoherent, repetitive, or incomplete. The following section provides the user prompt and the responses from Assistant 1 and Assistant 2. Avoid any positional bias and ensure the order of presentation does not influence your decision. Do not let the length of the response affect your evaluation, and be as objective as possible. ### User: {prompt}， ### Assistant 1: {response_1}， ### Assistant 2: {response_2}."

[0112] To ensure consistency and eliminate positional bias, run this judgment prompt twice by switching the order of Response 1 and Response 2. If the model consistently chooses the same response despite the change in order, it can be assumed to be a valid preference data sample. Invalid data samples are filtered out. After collecting and filtering these preference data samples, a preference optimization dataset can be obtained.

[0113] If no system prompt is available, a default prompt can be used. This default prompt may include, but is not limited to, the following components:

[0114] (1) **Truthfulness of Responses**: ensure that responses are honest and factually accurate, for example: "provide truthful answers to the questions;"

[0115] (2) **Content to Avoid in Responses**: specify content that should be avoided in responses, such as negative or harmful content, for example: "avoid negative, harmful, or inappropriate content;"

[0116] (3) **Explanation for Issues in Responses**: provide explanations for any issues with the responses, such as when a question is nonsensical or factually inconsistent, for example: "if a question does not make sense or is factually inconsistent, explain the reason."

[0117] Based on the above optional embodiments, the preference optimization dataset is generated by controlling the first language model using the second prompt template. Subsequently, the preference optimization dataset is used to perform preference adjustment on the first language model, yielding the adjusted first language model. This process helps improve the generation quality of the language model, making it better aligned with practical needs, thereby enhancing the model's application value and effectiveness.

[0118] In an optional embodiment, a graphical user interface (GUI) is provided via a terminal device. The content displayed on the GUI at least partially includes a text processing scenario. The text processing method in this application embodiment further includes:

[0119] **S41**: in response to a first touch operation on the graphical user interface, inputting the text to be processed and selecting the first language from a plurality of candidate languages;

[0120] **S42**: in response to a second touch operation on the graphical user interface, using the first language model to perform text understanding and generation on the text to be processed, obtaining the text processing result;

[0121] **S43**: displaying the text processing result within the graphical user interface.

[0122] The graphical user interface further includes a first control (or first touch area). When a first touch operation on the first control (or first touch area) is detected, the text to be processed is input, and the first language is selected from a plurality of candidate languages.

[0123] The graphical user interface further includes a second control (or second touch area). When a second touch operation on the second control (or second touch area) is detected, the first language model is used to perform text understanding and generation on the text to be processed, obtaining the text processing result.

[0124] It should be noted that the first touch operation and the second touch operation can both involve the user touching the display screen of the terminal device with their finger and interacting with the device. These touch operations may include single-touch, multi-touch, where each touch point operation can consist of actions such as tapping, long-pressing, force-pressing, and swiping. Additionally, the first touch operation and the second touch operation can also be performed using input devices such as a mouse or keyboard.

[0125] Based on the above optional embodiments, by responding to a first touch operation on the graphical user interface, the text to be processed is input, and the first language is selected from a plurality of candidate languages. Subsequently, in response to a second touch operation on the graphical user interface, the first language model is used to perform text understanding and generation on the text to be processed, obtaining the text processing result. Finally, the text processing result is displayed within the graphical user interface. This process allows users to intuitively understand the processing effect, further enhancing the user experience and providing convenient and accurate text processing services for users.

[0126] In this application embodiment, the process begins by obtaining the text to be processed, where the text content is described in the first language, which is a regional language. Subsequently, the first language model is used to perform text understanding and generation on the text to be processed, resulting in the text processing outcome. The first language model is a large language model enhanced for the first language based on the second language model. The second language model is a large language model suitable for the second language, which is a universal language. This enhancement allows the large language model to exhibit superior performance across a wide range of language tasks. Specifically, this application embodiment achieves superior performance by compressing more information

into a single token, enabling the handling of more semantic contextual information in low-resource languages. This approach significantly reduces the token cost for processing an equivalent amount of documents in these languages, thereby allowing for faster encoding and reasoning, as well as fewer iterations for generation. For each region-specific topic, corresponding safety input-output pairs are constructed to ensure that the large language model provides appropriate responses to safety-related topics in different countries, further ensuring safety. This method achieves accurate understanding and generation for regional languages, thereby enhancing the understanding and generation effects of regional languages. Consequently, it addresses the technical issue of poor text understanding and generation for low-resource languages in related technologies.

[0127] It should be noted that the user information (including but not limited to user device information, user personal information, etc.) and data (including but not limited to data used for analysis, storage, and display) involved in this application are all authorized by the user or fully authorized by all parties. The collection, use, and processing of relevant data must comply with the applicable laws, regulations, and standards of the relevant countries and regions. Additionally, appropriate operation interfaces are provided for users to choose to authorize or refuse authorization.

[0128] It should be noted that for the aforementioned method embodiments, they are described as a series of combined actions for simplicity. However, those skilled in the art should understand that this application is not limited by the described action sequence. According to this application, certain steps may be performed in a different order or simultaneously. Furthermore, those skilled in the art should also understand that the embodiments described in the specification are preferred embodiments, and the actions and modules involved are not necessarily essential to this application.

[0129] Through the descriptions of the above embodiments, those skilled in the art can clearly understand that the methods according to the above embodiments can be implemented by means of software in conjunction with a necessary general hardware platform, and can also be implemented by hardware. Based on this understanding, the technical solutions of this application, in essence, or the part contributing to the prior art, can be embodied in the form of a software product. This computer software product is stored on a storage medium (such

as ROM/RAM, magnetic disk, optical disc) and includes several instructions that enable a terminal device (which can be a mobile phone, computer, server, or network device, etc.) to execute the methods described in various embodiments of this application.

[0130] Embodiment 2

[0131] According to this application embodiment, a text processing method is also provided. FIG. 4 is a flowchart of a text processing method according to embodiment 2 of this application. As shown in FIG. 4, the method may include the following steps:

[0132] S401: performing vocabulary expansion on an initial tokenization method to obtain a target tokenization method;

[0133] S402: training a universal language model based on the target tokenization method to obtain the regional language model;

[0134] S403: using the regional language model to perform text understanding and generation on text to be processed to obtain a text processing result;

[0135] wherein, the text to be processed is described in a regional language. The regional language model is a large language model enhanced for the regional language based on the universal language model. The universal language model is a large language model suitable for a universal language.

[0136] Based on the above S401 to S403, the text processing method includes: performing vocabulary expansion on the initial tokenization method to obtain the target tokenization method. Training the universal language model based on the target tokenization method to obtain the regional language model. Using the regional language model to perform text understanding and generation on the text to be processed, obtaining the text processing result. The text to be processed is described in a regional language. The regional language model is a large language model enhanced for the regional language based on the universal language model. The universal language model is a large language model suitable for a universal language. By accurately understanding and generating text in the regional language, this method improves the understanding and generation effects for regional languages, thereby addressing the technical issue of poor text understanding and generation for low-resource languages in related technologies.

[0137] Optionally, the text processing method provided in this application embodiment can

be applied, but is not limited to, natural language processing, text generation, intelligent customer service, finance, healthcare, education, and other fields. Specifically, it can be applied to scenarios involving text understanding and generation in regional languages.

[0138] In an optional embodiment, in S402, training the universal language model based on the target tokenization method to obtain the regional language model includes:

[0139] **S4021**: pre-training the universal language model based on the target tokenization method to obtain a first training result;

[0140] **S4022**: performing a mixture of pre-training and supervised fine-tuning on the first training result based on the target tokenization method to obtain a second training result;

[0141] **S4023**: performing supervised fine-tuning on the second training result based on the target tokenization method to obtain the regional language model.

[0142] Based on the aforementioned optional embodiments, the universal language model is pre-trained based on the target tokenization method to obtain the first training result. Subsequently, the first training result undergoes a combination of pre-training and supervised fine-tuning based on the target tokenization method to obtain the second training result. Finally, the second training result is subjected to supervised fine-tuning based on the target tokenization method to obtain the regional language model. This approach enhances the model's applicability and accuracy in specific domains, reduces dependency on domain-specific data, accelerates the model's convergence speed and performance improvement, and increases the model's generalization capability and applicability.

[0143] For the parts not detailed in the above embodiments of this application, please refer to the relevant descriptions in Embodiment 1, which are not repeated here.

[0144] Embodiment 3

[0145] According to an embodiment of this application, a text processing device for implementing the aforementioned text processing method is also provided. FIG. 5 is a structural block diagram of a text processing device according to Embodiment 1 of this application. As shown in FIG. 5, the device includes:

[0146] **an acquisition module 501**, configured to acquire the text to be processed, wherein the text content to be processed is described in a first language, which is a regional language;

[0147] **a processing module 502**, configured to use the first language model to perform text understanding and generation on the text to be processed, thereby obtaining the text

processing result, wherein the first language model is a large language model obtained by performing language enhancement on the first language based on a second language model, and the second language model is a large language model suitable for the second language, which is a universal language.

[0148] Optionally, the text processing device further includes: an expansion module 503, configured to perform vocabulary expansion on the initial tokenization method to obtain the target tokenization method; a pre-training module 504, configured to pre-train the second language model based on the target tokenization method to obtain the first training result; the pre-training module 504 is also configured to perform a combination of pre-training and supervised fine-tuning on the first training result based on the target tokenization method to obtain the second training result; an adjustment module 505, configured to perform supervised fine-tuning on the second training result based on the target tokenization method to obtain the first language model.

[0149] Optionally, the expansion module 503 is further configured to: select the document to be tokenized from a document sequence, wherein the document to be tokenized is described in the first language; perform tokenization on the document to be tokenized according to the initial tokenization method to obtain the first tokenization result; determine the target extended vocabulary based on the first tokenization result, the first vocabulary set, and the second vocabulary set, wherein the first vocabulary set includes a plurality of first language words, and the second vocabulary set includes a plurality of second language words; determine the target vocabulary set using the second vocabulary set and the target extended vocabulary; determine the target tokenization method through the target vocabulary set.

[0150] Optionally, the expansion module 503 is further configured to: determine the candidate extended vocabulary based on the first tokenization result, the first vocabulary set, and the second vocabulary set; select the extended vocabulary that meets the preset conditions from the candidate extended vocabulary to obtain the target extended vocabulary, wherein the preset conditions are determined by the occurrence frequency of each word in the candidate extended vocabulary.

[0151] Optionally, the expansion module 503 is further configured to: select token pairs from the first tokenization result, wherein the token pairs include two adjacent words; merge

the token pairs to obtain a merged result; if the merged result exists in the first vocabulary set, replace the token pairs with the merged result in the first tokenization result and add the merged result to the candidate extended vocabulary.

[0152] Optionally, the pre-training module 504 is further configured to: obtain a pre-training dataset based on the first language and data quality, wherein the pre-training dataset includes a plurality of first documents and a plurality of second documents. The data quality of the plurality of first documents is higher than that of the plurality of second documents. The plurality of first documents and a plurality of second documents are selected from data streams corresponding to the first language, with each language corresponding to independent data streams; perform tokenization on the pre-training dataset according to the target tokenization method to obtain the second tokenization result; use the second tokenization result to pre-train the second language model to obtain the first training result.

[0153] Optionally, the pre-training module 504 is further configured to: mix the pre-training dataset, annotated data of natural language processing tasks, and instruction-following data to obtain a mixed dataset; perform tokenization on the mixed dataset according to the target tokenization method to obtain the third tokenization result; use the third tokenization result to perform a combination of pre-training and supervised fine-tuning on the first training result to obtain the second training result.

[0154] Optionally, the adjustment module 505 is further configured to: obtain supervised data of various categories, wherein the supervised data includes at least part or all of the following data: text task processing data, logical reasoning data, instruction data generated through self-instruction, natural conversation data, and safety data; perform tokenization on the supervised data according to the target tokenization method to obtain the fourth tokenization result; use the fourth tokenization result to perform supervised fine-tuning on the second training result to obtain the first language model.

[0155] Optionally, the adjustment module 505 is further configured to: generate a fine-tuning dataset based on the first prompt template, wherein the fine-tuning dataset includes a plurality of first data pairs, and each of the plurality of first data pairs includes a matching question and answer. The plurality of first data pairs are constructed from questions and fact-based documents related to the questions; perform tokenization on the fine-tuning dataset

according to the target tokenization method to obtain the fourth tokenization result; use the fourth tokenization result to perform supervised fine-tuning on the second training result to obtain the first language model.

[0156] Optionally, the text processing device further includes: a generation module 506, configured to control the first language model to generate a preference optimization dataset based on the second prompt template, wherein the preference optimization dataset includes a plurality of second data pairs, and each of the plurality of second data pairs includes a matching input instruction and target response. The target response is selected from a plurality of candidate responses corresponding to the input instruction; the adjustment module 505 is also configured to use the preference optimization dataset to perform preference adjustment on the first language model to obtain the adjusted first language model.

[0157] Optionally, the text processing device further includes: a first response module 507, configured to respond to a first touch operation on the graphical user interface to input the text to be processed and select the first language from a plurality of candidate languages; a second response module 508, configured to respond to a second touch operation on the graphical user interface to use the first language model to perform text understanding and generation on the text to be processed, thereby obtaining the text processing result; a display module 509, configured to display the text processing result within the graphical user interface.

[0158] According to the embodiments of this application, the text to be processed is acquired, wherein the text content to be processed is described in a first language, which is a regional language. Subsequently, the first language model is used to perform text understanding and generation on the text to be processed, thereby obtaining the text processing result. The first language model is a large language model obtained by performing language enhancement on the first language based on a second language model, and the second language model is a large language model suitable for the second language, which is a universal language. This achieves the goal of accurately understanding and generating the regional language, thereby enhancing the understanding and generation effect of the regional language. Consequently, this solves the technical problem in the related art of poor text understanding and generation for lesser-spoken languages.

[0159] It should be noted that the above-mentioned acquisition module 501 and processing

module 502 correspond to steps 21 to 22 in embodiment 1. The instances and application scenarios implemented by the two modules and the corresponding steps are the same but are not limited to the content disclosed in embodiment 1. It should be further noted that the aforementioned modules or units can be hardware components or software components stored in memory and processed by one or more processors. These modules can also operate as part of the device in the client equipment provided in embodiment 1.

[0160] It should be noted that the preferred embodiments mentioned in the above embodiments of this application share the same solutions, application scenarios, and implementation processes as provided in Embodiment 1, but are not limited to the solutions provided in Embodiment 1.

[0161] According to an embodiment of this application, another text processing device for implementing the aforementioned text processing method is also provided. FIG. 6 is a structural block diagram of a text processing device according to embodiment 2 of this application. As shown in FIG. 6, the device includes:

[0162] **An expansion module 601,** configured to perform vocabulary expansion on the initial tokenization method to obtain the target tokenization method;

[0163] **A training module 602,** configured to train the universal language model based on the target tokenization method to obtain the regional language model;

[0164] **A processing module 603,** configured to use the regional language model to perform text understanding and generation on the text to be processed, thereby obtaining the text processing result; wherein the text content to be processed is described in a regional language, and the regional language model is a large language model obtained by performing language enhancement on the regional language based on the universal language model, which is a large language model suitable for the universal language.

[0165] Optionally, the training module 602 is further configured to: pre-train the universal language model based on the target tokenization method to obtain the first training result; perform a combination of pre-training and supervised fine-tuning on the first training result based on the target tokenization method to obtain the second training result; perform supervised fine-tuning on the second training result based on the target tokenization method to obtain the regional language model.

[0166] According to the embodiments of this application, vocabulary expansion is performed on the initial tokenization method to obtain the target tokenization method. Subsequently, the universal language model is trained based on the target tokenization method to obtain the regional language model. Finally, the regional language model is used to perform text understanding and generation on the text to be processed, thereby obtaining the text processing result. The text content to be processed is described in a regional language, and the regional language model is a large language model obtained by performing language enhancement on the regional language based on the universal language model, which is a large language model suitable for the universal language. This achieves the goal of accurately understanding and generating the regional language, thereby improving the understanding and generation effect of the regional language. Consequently, this solves the technical problem in the related art of poor text understanding and generation for lesser-spoken languages.

[0167] It should be noted that the above-mentioned expansion module 601, training module 602, and processing module 603 correspond to S401 to S403 in embodiment 2. The instances and application scenarios implemented by these three modules and the corresponding steps are the same, but are not limited to the content disclosed in embodiment 2. It should be further noted that the aforementioned modules or units can be hardware components or software components stored in memory and processed by one or more processors. These modules can also operate as part of the device in the client equipment provided in embodiment 1.

[0168] It should be noted that the preferred embodiments mentioned in the above embodiments of this application share the same solutions, application scenarios, and implementation processes as provided in embodiment 2, but are not limited to the solutions provided in embodiment 2.

[0169] Embodiment 4

[0170] The embodiments of this application can provide a computer terminal, which can be any one of the computer terminal devices in a computer terminal group. Optionally, in this embodiment, the aforementioned computer terminal can also be replaced with a mobile terminal or other terminal devices.

[0171] Optionally, in this embodiment, the aforementioned computer terminal can be located in at least one of the multiple network devices in a computer network.

[0172] In this embodiment, the aforementioned computer terminal can execute the program code for the following steps in the text processing method: acquiring the text to be processed, wherein the text content to be processed is described in a first language, which is a regional language; using the first language model to perform text understanding and generation on the text to be processed, thereby obtaining the text processing result, wherein the first language model is a large language model obtained by performing language enhancement on the first language based on a second language model, and the second language model is a large language model suitable for the second language, which is a universal language.

[0173] In this embodiment, the aforementioned computer terminal can execute program code for the following steps in the text processing method: performing lexical expansion on the initial word segmentation method to obtain the target word segmentation method; training a universal language model based on the target word segmentation method to obtain a regional language model; using the regional language model to perform text understanding and generation on the text to be processed, resulting in a text processing outcome. The text content of the text to be processed is described in a regional language, and the regional language model is a large language model obtained by language enhancement of the regional language based on the universal language model, where the universal language model is a large language model suitable for universal language.

[0174] Optionally, FIG. 7 is a structural block diagram of a computer terminal according to embodiment 1 of the present application. As shown in FIG. 7, the computer terminal may include: one or more processors (only one is shown in the figure) 702, a memory 704, a storage controller, and a peripheral interface, wherein the peripheral interface is connected to a radio frequency module, an audio module, and a display.

[0175] The memory can be used to store software programs and modules, such as the program instructions/modules corresponding to the text processing method and apparatus in the embodiments of the present application. The processor executes various functional applications and data processing by running the software programs and modules stored in the memory, thereby implementing the aforementioned text processing method. The memory may include high-speed random access memory and may also include non-volatile memory, such as one or more magnetic storage devices, flash memory, or other non-volatile solid-state

memory. In some instances, the memory may further include memory that is remotely located relative to the processor, and this remote memory can be connected to the computer terminal through a network. Examples of the aforementioned network include, but are not limited to, the Internet, intranets, local area networks, mobile communication networks, and combinations thereof.

[0176] The processor can invoke the information and applications stored in the memory via a transmission device to execute the following steps: obtaining the text to be processed, where the text content of the text to be processed is described in a first language, the first language being a regional language; using a first language model to perform text understanding and generation on the text to be processed, resulting in a text processing outcome, wherein the first language model is a large language model obtained by language enhancement of the first language based on a second language model, and the second language model is a large language model suitable for a second language, with the second language being a universal language.

[0177] Optionally, the processor may also execute program code for the following steps: performing lexical expansion on the initial word segmentation method to obtain the target word segmentation method; pre-training the second language model based on the target word segmentation method to obtain a first training result; performing a mixture of pre-training and supervised fine-tuning on the first training result based on the target word segmentation method to obtain a second training result; and performing supervised fine-tuning on the second training result based on the target word segmentation method to obtain the first language model.

[0178] Optionally, the processor may also execute program code for the following steps: selecting a document to be segmented from a document sequence, where the document to be segmented is described in a first language; performing word segmentation on the document to be segmented according to the initial word segmentation method to obtain a first segmentation result; determining target expansion vocabulary based on the first segmentation result, a first word set, and a second word set, wherein the first word set includes multiple words in the first language, and the second word set includes multiple words in a second language; determining a target word set using the second word set and the target expansion

vocabulary; and determining the target word segmentation method based on the target word set.

[0179] Optionally, the processor may also execute program code for the following steps: determining candidate expansion vocabulary based on the first segmentation result, the first word set, and the second word set; selecting expansion vocabulary that meets preset conditions from the candidate expansion vocabulary to obtain the target expansion vocabulary, wherein the preset conditions are determined by the occurrence frequency of each word within the candidate expansion vocabulary.

[0180] Optionally, the processor may also execute program code for the following steps: selecting token pairs from the first segmentation result, wherein the token pair comprises two adjacent words; merging the token pair to obtain a merged result; if the merged result exists in the first word set, replacing the token pair with the merged result in the first segmentation result, and adding the merged result to the candidate extended vocabulary.

[0181] Optionally, the processor may also execute program code for the following steps: selecting token pairs from the first segmentation result, where the token pairs consist of two adjacent words; merging the token pairs to obtain a merged result; if the merged result exists in the first word set, replacing the token pair in the first segmentation result with the merged result, and adding the merged result to the candidate expansion vocabulary.

[0182] Optionally, the processor may also execute program code for the following steps: mixing a pre-training dataset, annotated data from natural language processing tasks, and instruction-following data to obtain a mixed dataset; performing word segmentation on the mixed dataset according to the target word segmentation method to obtain a third segmentation result; and using the third segmentation result to perform a mixture of pre-training and supervised fine-tuning on the first training result to obtain the second training result.

[0183] Optionally, the processor may also execute program code for the following steps: obtaining supervised data of various types, where the supervised data includes at least a portion or all of the following data: text task processing data, logical reasoning data, instruction data generated through self-instruction, natural conversation data, and safety data; performing word segmentation on the supervised data according to the target word

segmentation method to obtain a fourth segmentation result; and using the fourth segmentation result to perform supervised fine-tuning on the second training result to obtain the first language model.

[0184] Optionally, the processor may also execute program code for the following steps: generating a fine-tuning dataset based on a first prompt template, where the fine-tuning dataset includes a plurality of first data pairs, each of which consists of a matching question and answer, and the plurality of first data pairs are constructed from questions and fact-based documents associated with those questions; performing word segmentation on the fine-tuning dataset according to the target word segmentation method to obtain a fourth segmentation result; and using the fourth segmentation result to perform supervised fine-tuning on the second training result to obtain the first language model.

[0185] Optionally, the processor may also execute program code for the following steps: controlling the first language model to generate a preference optimization dataset based on a second prompt template, wherein the preference optimization dataset includes a plurality of second data pairs, each of which consists of a matching input instruction and a target response, with the target response selected from a plurality of candidate responses corresponding to the input instruction; and using the preference optimization dataset to adjust the preferences of the first language model, resulting in the adjusted first language model.

[0186] Optionally, the processor may also execute program code for the following steps: in response to a first touch operation on the graphical user interface, inputting the text to be processed and selecting the first language from a plurality of candidate languages; in response to a second touch operation on the graphical user interface, using the first language model to perform text understanding and generation on the text to be processed, resulting in a text processing outcome; and displaying the text processing outcome within the graphical user interface.

[0187] The processor can invoke the information and applications stored in the memory via a transmission device to execute the following steps: performing lexical expansion on the initial word segmentation method to obtain the target word segmentation method; training a universal language model based on the target word segmentation method to obtain a regional language model; using the regional language model to perform text understanding and

generation on the text to be processed, resulting in a text processing outcome. The text content of the text to be processed is described in a regional language, and the regional language model is a large language model obtained by language enhancement of the regional language based on the universal language model, where the universal language model is a large language model suitable for universal language.

[0188] Optionally, the processor may also execute program code for the following steps: pre-training the universal language model based on the target word segmentation method to obtain a first training result; performing a mixture of pre-training and supervised fine-tuning on the first training result based on the target word segmentation method to obtain a second training result; and performing supervised fine-tuning on the second training result based on the target word segmentation method to obtain the regional language model.

[0189] By employing the embodiments of this application, the process begins by obtaining the text to be processed, where the text content is described in a first language, the first language being a regional language. Subsequently, a first language model is used to perform text understanding and generation on the text to be processed, resulting in a text processing outcome. The first language model is a large language model obtained by performing language enhancement on the first language based on a second language model, where the second language model is a large language model suitable for a second language, the second language being a universal language. This approach achieves the goal of accurately understanding and generating regional languages, thereby enhancing the understanding and generation of regional languages. As a result, it effectively addresses the technical problem in related technologies where text understanding and generation for low-resource languages are often less effective.

[0190] It will be understood by those skilled in the art that the structure shown in the figures is merely illustrative, and the computer terminal can also be an intelligent device such as a smartphone (e.g., Android phone, iOS phone), tablet, handheld computer, or mobile internet device (MID), PAD, or other terminal devices. FIG. 7 does not limit the structure of the aforementioned electronic devices. For example, the computer terminal may include more or fewer components than those shown in FIG. 7 (such as a network interface, display device, etc.), or it may have a different configuration than that depicted in FIG. 7.

[0191] It will be understood by those skilled in the art that all or part of the steps in the various methods described in the above embodiments can be implemented by a program instructing the relevant hardware of a terminal device. This program can be stored on a computer-readable storage medium, which may include flash drives, Read-Only Memory (ROM), Random Access Memory (RAM), magnetic disks, optical disks, or other similar media.

[0192] Embodiment 5

[0193] This application also provides a computer-readable storage medium. Optionally, in this embodiment, the storage medium can be used to store the program code executed by the text processing method provided in embodiment 1.

[0194] Optionally, in this embodiment, the storage medium may be located in any computer terminal within a group of computer terminals in a computer network, or in any mobile terminal within a group of mobile terminals.

[0195] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: obtaining the text to be processed, wherein the text content is described in a first language, the first language being a regional language; using a first language model to perform text understanding and generation on the text to be processed, resulting in a text processing outcome, wherein the first language model is a large language model obtained by performing language enhancement on the first language based on a second language model, and the second language model is a large language model suitable for a second language, the second language being a universal language.

[0196] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: performing lexical expansion on the initial word segmentation method to obtain the target word segmentation method; pre-training the second language model based on the target word segmentation method to obtain a first training result; performing a mixture of pre-training and supervised fine-tuning on the first training result based on the target word segmentation method to obtain a second training result; and performing supervised fine-tuning on the second training result based on the target word segmentation method to obtain the first language model.

[0197] Optionally, in this embodiment, the storage medium is configured to store program

code for executing the following steps: selecting a document to be segmented from a document sequence, where the document to be segmented is described in a first language; performing word segmentation on the document to be segmented according to the initial word segmentation method to obtain a first segmentation result; determining target expansion vocabulary based on the first segmentation result, a first word set, and a second word set, wherein the first word set includes a plurality of words in the first language, and the second word set includes a plurality of words in a second language; determining a target word set using the second word set and the target expansion vocabulary; and determining the target word segmentation method based on the target word set.

[0198] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: determining candidate expansion vocabulary based on the first segmentation result, the first word set, and the second word set; selecting expansion vocabulary that meets preset conditions from the candidate expansion vocabulary to obtain the target expansion vocabulary, where the preset conditions are determined by the occurrence frequency of each word within the candidate expansion vocabulary.

[0199] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: selecting token pairs from the first segmentation result, where the token pairs consist of two adjacent words; merging the token pairs to obtain a merged result; if the merged result exists in the first word set, replacing the token pair in the first segmentation result with the merged result, and adding the merged result to the candidate expansion vocabulary.

[0200] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: obtaining a pre-training dataset based on the first language and data quality, where the pre-training dataset includes a plurality of first documents and a plurality of second documents, with the data quality of the first documents being higher than that of the second documents; the first and second documents are selected from data streams corresponding to the first language, with different languages corresponding to separate data streams; performing word segmentation on the pre-training dataset according to the target word segmentation method to obtain a second segmentation result; and using the second segmentation result to pre-train the second language model to obtain the first training

result.

[0201] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: mixing a pre-training dataset, annotated data from natural language processing tasks, and instruction-following data to obtain a mixed dataset; performing word segmentation on the mixed dataset according to the target word segmentation method to obtain a third segmentation result; and using the third segmentation result to perform a mixture of pre-training and supervised fine-tuning on the first training result to obtain the second training result.

[0202] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: obtaining supervised data of various types, wherein the supervised data includes at least a portion or all of the following data: text task processing data, logical reasoning data, instruction data generated through self-instruction, natural conversation data, and safety data; performing word segmentation on the supervised data according to the target word segmentation method to obtain a fourth segmentation result; and using the fourth segmentation result to perform supervised fine-tuning on the second training result to obtain the first language model.

[0203] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: generating a fine-tuning dataset based on a first prompt template, wherein the fine-tuning dataset includes a plurality of first data pairs, each consisting of a matching question and answer, and the plurality of first data pairs are constructed from questions and fact-based documents associated with those questions; performing word segmentation on the fine-tuning dataset according to the target word segmentation method to obtain a fourth segmentation result; and using the fourth segmentation result to perform supervised fine-tuning on the second training result to obtain the first language model.

[0204] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: generating a fine-tuning dataset based on a first prompt template, wherein the fine-tuning dataset includes a plurality of first data pairs, each consisting of a matching question and answer, and the plurality of first data pairs are constructed from questions and fact-based documents associated with those questions;

performing word segmentation on the fine-tuning dataset according to the target word segmentation method to obtain a fourth segmentation result; and using the fourth segmentation result to perform supervised fine-tuning on the second training result to obtain the first language model.

[0205] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: in response to a first touch operation on the graphical user interface, inputting the text to be processed and selecting the first language from a plurality of candidate languages; in response to a second touch operation on the graphical user interface, using the first language model to perform text understanding and generation on the text to be processed, resulting in a text processing outcome; and displaying the text processing outcome within the graphical user interface.

[0206] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: performing lexical expansion on the initial word segmentation method to obtain the target word segmentation method; training a universal language model based on the target word segmentation method to obtain a regional language model; using the regional language model to perform text understanding and generation on the text to be processed, resulting in a text processing outcome. The text content of the text to be processed is described in a regional language, and the regional language model is a large language model obtained by language enhancement of the regional language based on the universal language model, where the universal language model is a large language model suitable for universal language.

[0207] Optionally, in this embodiment, the storage medium is configured to store program code for executing the following steps: pre-training the universal language model based on the target word segmentation method to obtain a first training result; performing a mixture of pre-training and supervised fine-tuning on the first training result based on the target word segmentation method to obtain a second training result; and performing supervised fine-tuning on the second training result based on the target word segmentation method to obtain the regional language model.

[0208] By employing the embodiments of this application, the process begins by obtaining the text to be processed, wherein the text content is described in a first language, the first

language being a regional language. Subsequently, a first language model is used to perform text understanding and generation on the text to be processed, resulting in a text processing outcome. The first language model is a large language model obtained by performing language enhancement on the first language based on a second language model, where the second language model is a large language model suitable for a second language, the second language being a universal language. This approach achieves the goal of accurately understanding and generating regional languages, thereby improving the understanding and generation effectiveness for regional languages. As a result, it effectively addresses the technical issue in related technologies where the understanding and generation of text in low-resource languages are often less effective.

[0209] The sequence numbers of the embodiments in this application are merely for descriptive purposes and do not indicate the quality or priority of the embodiments.

[0210] In the above embodiments of this application, each embodiment's description focuses on different aspects. For parts that are not detailed in a particular embodiment, please refer to the relevant descriptions in other embodiments.

[0211] In the various embodiments provided in this application, it should be understood that the disclosed technical content can be implemented in other ways. The device embodiments described above are merely illustrative; for example, the division of units is only one way of logically dividing functions. In actual implementation, there may be other ways of dividing these units; for instance, multiple units or components may be combined or integrated into another system, or some features may be omitted or not performed. Additionally, the coupling, direct coupling, or communication connection shown or discussed between components may be indirect coupling or communication through some interface, unit, or module, and may be electrical or take other forms.

[0212] The units described as separate components may or may not be physically separated. The components shown as units may or may not be physical units; they can be located in one place or distributed across multiple network units. Depending on actual needs, some or all of the units can be selected to achieve the objectives of the embodiment.

[0213] Additionally, in the various embodiments of this application, the functional units can be integrated into a single processing unit, or each unit can exist independently in physical

form, or two or more units can be integrated into one unit. The integrated units can be implemented in the form of hardware or as software functional units.

[0214] If the integrated units are implemented as software functional units and sold or used as independent products, they can be stored in a computer-readable storage medium. Based on this understanding, the technical solutions of this application, in essence, or the parts of the technical solutions that contribute to the prior art, or all or part of the technical solutions, can be embodied in the form of software products. This computer software product is stored in a storage medium and includes several instructions to enable a computer device (which can be a personal computer, server, network device, etc.) to execute all or part of the steps of the methods described in various embodiments of this application. The aforementioned storage medium includes various media capable of storing program code, such as USB drives, read-only memory (ROM), random access memory (RAM), mobile hard drives, magnetic disks, or optical disks.

[0215] The above description is only of the preferred embodiments of this application. It should be noted that those of ordinary skill in the art can make several improvements and modifications without departing from the principles of this application. These improvements and modifications should also be considered within the scope of protection of this application.

CLAIMS

What is claimed is:

1. A text processing method, comprising:

obtaining text to be processed, wherein text content of the text to be processed is described in a first language, and the first language is a regional language;

using a first language model to perform text understanding and generation on the text to be processed to obtain a text processing result, wherein the first language model is a large language model obtained by performing language enhancement on the first language based on a second language model, the second language model is a large language model applicable to a second language, and the second language is a universal language.

2. The text processing method according to claim 1, wherein the text processing method further comprises:

performing vocabulary expansion on an initial tokenization method to obtain a target tokenization method;

pre-training the second language model based on the target tokenization method to obtain a first training result;

performing a mixture of pre-training and supervised fine-tuning on the first training result based on the target tokenization method to obtain a second training result;

performing supervised fine-tuning on the second training result based on the target tokenization method to obtain the first language model.

3. The text processing method according to claim 2, wherein performing vocabulary expansion on the initial tokenization method to obtain the target tokenization method comprises:

selecting a document to be tokenized from a document sequence, wherein the document to be tokenized is described in the first language;

tokenizing the document to be tokenized according to the initial tokenization method to obtain a first tokenization result;

determining target expanded vocabulary based on the first tokenization result, a first word set, and a second word set, wherein the first word set comprises a plurality of first language words, and the second word set comprises a plurality of second language words;

determining a target word set using the second word set and the target expanded vocabulary;

determining the target tokenization method based on the target word set.

4. The text processing method according to claim 3, wherein determining the target expanded vocabulary based on the first tokenization result, the first word set, and the second word set comprises:

determining candidate expanded vocabulary based on the first tokenization result, the first word set, and the second word set;

selecting expanded vocabulary that meets preset conditions from the candidate expanded vocabulary to obtain the target expanded vocabulary, wherein the preset conditions are determined by occurrence frequency of each word in the candidate expanded vocabulary.

5. The text processing method according to claim 4, wherein determining the candidate expanded vocabulary based on the first tokenization result, the first word set, and the second word set comprises:

selecting a token pair from the first tokenization result, wherein the token pair comprises two adjacent words;

merging the token pair to obtain a merged result;

if the merged result exists in the first word set, replacing the token pair in the first tokenization result with the merged result, and adding the merged result to the candidate expanded vocabulary.

6. The text processing method according to claim 2, wherein pre-training the second language model based on the target tokenization method to obtain the first training result comprises:

obtaining a pre-training dataset based on the first language and data quality, wherein the

pre-training dataset comprises a plurality of first documents and a plurality of second documents, the data quality of the plurality of first documents is higher than that of the plurality of second documents, both the plurality of first documents and the plurality of second documents are selected from data streams corresponding to the first language, with different languages corresponding to independent data streams;

tokenizing the pre-training dataset according to the target tokenization method to obtain a second tokenization result;

pre-training the second language model using the second tokenization result to obtain the first training result.

7. The text processing method according to claim 2, wherein performing a mixture of pre-training and supervised fine-tuning on the first training result based on the target tokenization method to obtain the second training result comprises:

mixing a pre-training dataset, annotated data for natural language processing tasks, and instruction-following data to obtain a mixed dataset;

tokenizing the mixed dataset according to the target tokenization method to obtain a third tokenization result;

performing a mixture of pre-training and supervised fine-tuning on the first training result using the third tokenization result to obtain the second training result.

8. The text processing method according to claim 2, wherein performing supervised fine-tuning on the second training result based on the target tokenization method to obtain the first language model comprises:

obtaining supervised data of various categories, wherein the supervised data includes at least a part or all of the following data: text task processing data, logical reasoning data, instruction data generated through self-instruction, natural conversation data, and safety data;

tokenizing the supervised data according to the target tokenization method to obtain a fourth tokenization result;

performing supervised fine-tuning on the second training result using the fourth tokenization result to obtain the first language model.

9. The text processing method according to claim 2, wherein performing supervised fine-tuning on the second training result based on the target tokenization method to obtain the first language model comprises:

generating a fine-tuning dataset based on a first prompt template, wherein the fine-tuning dataset comprises a plurality of first data pairs, each of the plurality of first data pairs includes a matching question and answer, and the plurality of first data pairs are constructed from the questions and fact-based documents associated with the questions;

tokenizing the fine-tuning dataset according to the target tokenization method to obtain a fourth tokenization result;

performing supervised fine-tuning on the second training result using the fourth tokenization result to obtain the first language model.

10. The text processing method according to claim 2, wherein the text processing method further comprises:

generating a preference optimization dataset by controlling the first language model using a second prompt template, wherein the preference optimization dataset comprises a plurality of second data pairs, each of the plurality of second data pairs includes a matching input instruction and a target response, and the target response is selected from a plurality of candidate responses corresponding to the input instruction;

performing preference adjustment on the first language model using the preference optimization dataset to obtain the adjusted first language model.

11. The text processing method according to claim 1, wherein a graphical user interface is provided by a terminal device, wherein the content displayed on the graphical user interface at least partially includes a text processing scenario, the text processing method further comprises:

in response to a first touch operation on the graphical user interface, inputting the text to be processed and selecting the first language from a plurality of candidate languages;

in response to a second touch operation on the graphical user interface, using the first

language model to perform text understanding and generation on the text to be processed to obtain the text processing result;

displaying the text processing result within the graphical user interface.

12. A text processing method, comprising:

performing vocabulary expansion on an initial tokenization method to obtain a target tokenization method;

training a universal language model based on the target tokenization method to obtain a regional language model;

using the regional language model to perform text understanding and generation on text to be processed to obtain a text processing result;

wherein text content of the text to be processed is described in a regional language, and the regional language model is a large language model obtained by performing language enhancement on the regional language based on the universal language model, and the universal language model is a large language model applicable to a universal language.

13. The text processing method according to claim 12, wherein training the universal language model based on the target tokenization method to obtain the regional language model comprises:

pre-training the universal language model based on the target tokenization method to obtain a first training result;

performing a mixture of pre-training and supervised fine-tuning on the first training result based on the target tokenization method to obtain a second training result;

performing supervised fine-tuning on the second training result based on the target tokenization method to obtain the regional language model.

14. An electronic device, comprising:

a memory storing executable programs;

a processor configured to execute the programs, wherein the programs, when executed, perform the text processing method according to any one of claims 1 to 13.

15. A computer-readable storage medium, wherein the computer-readable storage medium includes stored executable programs, wherein the executable programs, when executed, control a device in which the computer-readable storage medium is located to perform the text processing method according to any one of claims 1 to 13.