United States Patent Application Publication	20250266032
Kind Code	A1
Publication Date	August 21, 2025
Inventor(s)	ZHANG; Haitong et al.

METHOD AND DEVICE FOR GENERATING SPEECH, STORAGE MEDIUM, AND ELECTRONIC DEVICE

Abstract

A method for generating a speech includes acquiring a voice feature vector of a voice to be processed, and inputting the voice feature vector into a voice generation model to obtain a language unit vector; acquiring a text feature vector, and determining, according to the text feature vector and the language unit vector, a feature vector to be processed; and inputting the feature vector to be processed into a sequence-to-sequence model to obtain an acoustic feature vector, and inputting the acoustic feature vector into a vocoder to obtain a target voice corresponding to the voice to be processed or the text feature vector.

Inventors:	**ZHANG; Haitong (Hangzhou, Zhejiang, CN), LIN; Yue (Hangzhou, Zhejiang, CN)**
Applicant:	**NETEASE (HANGZHOU) NETWORK CO., LTD.** (Hangzhou, Zhejiang, CN)
Family ID:	**1000008589164**
Assignee:	**NETEASE (HANGZHOU) NETWORK CO., LTD. (Hangzhou, Zhejiang, CN)**
Appl. No.:	**18/858328**
Filed (or PCT Filed):	**September 22, 2022**
PCT No.:	**PCT/CN2022/120602**

Foreign Application Priority Data

CN	202210593870.9	May. 27, 2022

Publication Classification

Int. Cl.: **G10L13/027** (20130101); **G10L13/06** (20130101); **G10L13/08** (20130101); **G10L17/02** (20130101); **G10L19/00** (20130101); **G10L19/038** (20130101); **G10L25/18** (20130101)

U.S. Cl.:

CPC **G10L13/027** (20130101); **G10L13/06** (20130101); **G10L13/08** (20130101); **G10L17/02** (20130101); **G10L19/038** (20130101); **G10L25/18** (20130101); G10L2019/0002 (20130101); G10L2019/0013 (20130101)

Background/Summary

CROSS-REFERENCE TO RELATED APPLICAITON

[0001] The present disclosure claims the priority to the Chinese Patent Application NO. 202210593870.9, entitled “METHOD AND DEVICE FOR GENERATING SPEECH, STORAGE MEDIUM, AND ELECTRONIC DEVICE”, filed on May 27, 2022, the entire contents of which are incorporated herein by reference for all purposes.

TECHNICAL FIELD

[0002] The present disclosure relates to the field of speech processing technologies, and in particular to a method for generating a speech, a device for generating a speech, a non-transitory computer-readable storage medium, and an electronic device.

BACKGROUND

[0003] In recent years, with the rapid development of deep learning, Text to Speech (TTS) technologies have made significant progress. In addition, thanks to the development of various deep learning technologies, Voice Conversion (VC) has also made rapid progress.

[0004] It should be noted that the information disclosed in the background section above is only used to enhance the understanding of the background of the present disclosure, and therefore may include information that does not constitute the prior art known to those skilled in the art.

SUMMARY

[0005] According to a first aspect of embodiments of the present disclosure, there is provided a method for generating a speech, and the method includes. [0006] acquiring a speech feature vector of a to-be-processed speech, and inputting the speech feature vector into a speech generation model to obtain a language unit vector; [0007] acquiring a text feature vector, and determining a to-be-processed feature vector according to the text feature vector and the language unit vector; and [0008] inputting the to-be-processed feature vector into a sequence-to-sequence model to obtain an acoustic feature vector, and inputting the acoustic feature vector into a vocoder to obtain a target speech corresponding to the to-be-processed speech or the text feature vector.

[0009] According to a second aspect of embodiments of the present disclosure, there is provided an electronic device, including: a processor and a memory, wherein computer-readable instructions are stored in the memory, and the computer-readable instructions, when executed by the processor, implement the method for generating the speech in any of the above embodiments.

[0010] According to a third aspect of an embodiment of the present disclosure, there is provided a non-transitory computer-readable storage medium having a computer program stored thereon, wherein the computer program, when executed by a processor, implements the method for generating the speech in any of the above embodiments.

[0011] It should be understood that the above general description and the following detailed description are merely exemplary and explanatory, and should not limit the present disclosure.

Description

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] FIG. **1** schematically shows a flowchart of a method for generating a speech in an embodiment of the present disclosure;

[0013] FIG. **2** schematically shows a flowchart of a method for outputting a language unit vector by a speech generation model in an embodiment of the present disclosure;

[0014] FIG. **3** schematically shows a flowchart of a processing method by a speech generation model in an embodiment of the present disclosure;

[0015] FIG. **4** schematically shows a flowchart of a method for performing quantization to obtain a speech quantization sequence in an embodiment of the present disclosure;

[0016] FIG. **5** schematically shows a flowchart of a method for updating a codebook in an embodiment of the present disclosure;

[0017] FIG. **6** schematically shows a flowchart of a method for acquiring a speaker vector in an embodiment of the present disclosure;

[0018] FIG. **7** schematically shows a flowchart of a method for determining a to-be-processed feature vector according to a text feature vector and a language unit vector in an embodiment of the present disclosure;

[0019] FIG. **8** schematically shows a flowchart of a method for outputting an acoustic feature vector by a sequence-to-sequence model in an embodiment of the present disclosure;

[0020] FIG. **9** schematically shows a flowchart of a processing method by a sequence-to-sequence model in an embodiment of the present disclosure;

[0021] FIG. **10** schematically shows a flowchart of a method for outputting a target speech according to a generator in an embodiment of the present disclosure;

[0022] FIG. **11** schematically shows a flowchart of a method for performing loss calculation to obtain a third loss value in an embodiment of the present disclosure;

[0023] FIG. **12** schematically shows a schematic diagram of a framework of a speech generation model in an application scenario in an embodiment of the present disclosure;

[0024] FIG. **13** schematically shows a structural schematic diagram of a device for generating a speech in an embodiment of the present disclosure;

[0025] FIG. **14** schematically shows an electronic device for implementing a method for generating a speech in an embodiment of the present disclosure; and

[0026] FIG. **15** schematically shows a computer-readable storage medium for implementing a method for generating a speech in an embodiment of the present disclosure.

DETAILED DESCRIPTION

[0027] Example embodiments will now be described more fully with reference to the accompanying drawings. However, the example embodiments may be implemented in a variety of forms and should not be construed as being limited to examples set forth herein; rather, these embodiments are provided so that the present disclosure will be more complete and comprehensive so as to convey the spirit of the example embodiments to those skilled in this art. The described features, structures, or characteristics may be combined in any suitable manner in one or more embodiments. In the description below, numerous specific details are set forth to provide a thorough understanding of embodiments of the present disclosure. However, those skilled in the art will appreciate that the technical solution of the present disclosure may be practiced without one or more of the specific details, or other methods, components, devices, steps and the like may be employed. In other instances, well-known technical solutions are not shown or described in detail to avoid obscuring various aspects of the present disclosure.

[0028] The terms “one”, “a”, “the”, and “said” in the present specification are used to indicate that there are one or more elements/components or the like; the terms “include” and “have” are used to indicate an open meaning of including and means that there may be additional elements/components/etc. in addition to the listed elements/components/etc.; and the terms “first”, "second" etc. are used only as markers, and do not limit the number of objects.

[0029] In addition, the drawings are merely schematic representations of the present disclosure and are not necessarily drawn to scale. The same reference numerals in the drawings denote the same or similar parts, and the repeated description thereof will be omitted. Some of the block diagrams shown in the figures are functional entities and do not necessarily correspond to physically or logically separate entities.

[0030] In recent years, with the rapid development of deep learning, TTS technologies have made significant progress. In addition, thanks to the development of various deep learning technologies, VC has also made rapid progress. However, both TTS models and VC models require a large amount of recorded speech data (more than ten hours) to achieve the desired effect. Speech recording is very expensive and complicated.

[0031] Therefore, how to ensure the effect of TTS and VC when only a small amount of speaker data can be obtained has become a hot topic of research. This research is called speaker adaptation, or speaker cloning.

[0032] The speaker adaptation is a technology that quickly and automatically adapts a deep learning model to a target speaker, so that the performance of the deep learning model on this speaker is significantly improved.

[0033] From the perspective of the speaker cloning, the TTS technologies and the VC should be viewed as a system that generates a speech of the target speaker based on different inputs.

[0034] In the TTS field, the speaker adaptation can be divided into supervised speaker adaptation and unsupervised speaker adaptation.

[0035] The supervised speaker adaptation means that <text, speech> paired data is required during adaptation, while the unsupervised speaker adaptation means that only speech data is required during the adaptation without the corresponding text.

[0036] Previous researches have shown that in the supervised speaker adaptation, a small amount of <text, speech> paired data of the target speaker is used to fine-tune a multi-speaker base model, thereby achieving a high-quality effect.

[0037] However, in the unsupervised speaker adaptation, fine-tuning the model is not possible.

[0038] A common unsupervised speaker adaptation method uses a speaker recognition model to extract a speaker vector from a speech, and then uses this speaker vector to synthesize the speech of the speaker.

[0039] However, as the amount of data increases, the performance of this method will not be improved any further.

[0040] At present, some researches have combined the TTS technologies and the VC.

[0041] For example, a sequence-to-sequence TTS model is used to extract a speaker-independent representation to model a VC model; a TTS pre-trained model is used to improve the effect of the VC model; different encoders are used to encode different input source content, and then the same decoder is used for decoding, two tasks can be handled at the same time. However, the effect of TTS will always decline from the effect point of view.

[0042] Alternatively, different encoders are used to encode different input source content, but the training of the model is more complicated and requires more loss functions and hyperparameters. Therefore, the research combining the TTS technologies and the VC not only has a complex training method, but also cannot handle the TTS and the VC well enough to improve the performance of both tasks at the same time.

[0043] In view of problems existing in the related arts, the present disclosure provides a method for generating a speech. FIG. **1** shows a flowchart of a method for generating a speech. As shown in FIG. **1**, the method for generating the speech includes at least steps S**110** to S**130**.

[0044] In the step S**110**, a speech feature vector of a to-be-processed speech is acquired, and the speech feature vector is input into a speech generation model to obtain a language unit vector.

[0045] In the step S**120**, a text feature vector is acquired, and a to-be-processed feature vector is determined according to the text feature vector and the language unit vector.

[0046] In the step S**130**, the to-be-processed feature vector is input into a sequence-to-sequence model to obtain an acoustic feature vector, and the acoustic feature vector is input into a vocoder to obtain a target speech corresponding to the to-be-processed speech or the text feature vector.

[0047] In embodiments of the present disclosure, by acquiring the speech feature vector and the text feature vector, the speech and the text can be received as the inputs, which facilitate the fusion of a TTS task and a VC task to perform multimodal modeling, and improve the performance of the TTS task and the VC task. Furthermore, the speech feature vector and the text feature vector are acquired in the case of a small amount of data, which provides a variety of voice cloning strategies, improves the effect of voice cloning under the small amount of data, reduces the training difficulty and training duration of various models, and supports voice cloning methods in various application scenarios.

[0048] The following is a detailed description of each step of the method for generating the speech.

[0049] In the step S**110**, the speech feature vector of the to-be-processed speech is acquired, and the speech feature vector is input into the speech generation model to obtain the language unit vector.

[0050] In an embodiment of the present disclosure, the to-be-processed speech may be a speech that needs to be converted for voice conversion.

[0051] The voice conversion is a system that automatically converts a speech of speaker A into a speech of speaker B while keeping content of the speech unchanged.

[0052] Therefore, the to-be-processed speech may be understood as the speech of speaker A.

[0053] Correspondingly, the speech feature vector of the to-be-processed speech may be a Mel-spectrum feature vector extracted according to the to-be-processed speech.

[0054] Since the Mel spectrum feature simulates the human ear's processing characteristics for the speech to a certain extent, it can better reflect the human auditory characteristics, thereby improving the user's auditory experience.

[0055] After the speech feature vector is acquired, the speech feature vector may be input into the speech generation model, so that the speech generation model outputs a corresponding language unit vector.

[0056] In an embodiment, FIG. **2** shows a schematic flowchart of a method for outputting a language unit vector by a speech generation model. As shown in FIG. **2**, the method includes at least steps S**210** and S**220**. In the step S**210**, the speech feature vector is input into the speech generation model to make the speech generation model output a speech encoding vector and a self-restored speech.

[0057] In an embodiment, FIG. **3** shows a schematic flowchart of a processing method by a speech generation model. As shown in FIG. **3**, the method includes at least steps S**310** to S**330**. In the step S**310**, the speech feature vector is input into the speech generation model, and nonlinear transformation is performed, by using an encoder module of the speech generation model, on the speech feature vector to obtain the speech encoding vector.

[0058] The speech generation model may be a Vector Quatization-Variational AutoEncoder (VQ-VAE) model, or may also be another model, which are not particularly limited in the embodiments.

[0059] The VQ-VAE model is an autoencoder with an obvious feature that the encoding vector encoded by it is discrete. VQ-VAE includes an encoding layer and a decoding layer. The speech feature vector may be encoded into a discrete encoding vector through the encoding layer, and the discrete encoding vector may be decoded into a vector through the decoding layer.

[0060] Specifically, when the speech generation model is the VQ-VAE model, the speech feature vector is input into the speech generation model, and an encoder module in the VQ-VAE model maps the speech feature vector to a high-dimensional speech encoding vector through the nonlinear transformation. The speech encoding vector may be represented by Z.sub.1:N.

[0061] Through the nonlinear transformation of the neural network, the encoder can extract the more abstract and higher-dimensional feature than the input feature.

[0062] The encoder module in the VQ-VAE model may be composed of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM).

[0063] Specifically, the CNN model may include an input layer, a convolution layer, a pooling layer, a Fully Connected (FC) layer, and an output layer.

[0064] An activation function of the convolution layer is ReLU (which is a linear rectification function), while the pooling layer has no activation function. The combination of the convolution layer plus the pooling layer may appear many times in the hidden layer, and the number of times actually depends on the needs of the model.

[0065] Alternatively, the combination of the convolution layer plus the convolutional layer or the combination of the convolution layer plus the convolutional layer plus the pooling layer may also be flexibly used, which is not limited upon the building of the model. However, the most common CNN is several combinations of convolutional layers plus pooling layers.

[0066] The fully connected layer is after the several convolutional layers and pooling layers. The fully connected layer is actually a Deep Neural Networks (DNN) structure, but the output layer uses the Softmax activation function to perform classification and other tasks.

[0067] LSTM is a special Recurrent Neural Network (RNN), which is mainly used to solve the gradient vanishing and gradient exploding problems in a long sequence training process. In simple terms, compared with an ordinary RNN, LSTM can perform better in a longer sequence.

[0068] In the step S**320**, quantization is performed, by using a vector quantization module of the speech generation model, on the speech encoding vector to obtain a speech quantization sequence, and a speaker vector corresponding to the to-be-processed speech is acquired.

[0069] After the encoder module of the VQ-VAE model obtains the speech encoding vector Z.sub.1:N, the vector quantization ({circumflex over (z)}.sub.i=e.sub.k) module in the VQ-VAE model may be used to quantize the high-dimensional speech encoding vector into the speech quantization sequence. The speech quantization sequence may be represented by {circumflex over (z)}.sub.i:N.

[0070] In an embodiment, based on a codebook in the vector quantization module of the speech generation model, the quantization is performed on the speech encoding vector through a nearest neighbor search algorithm to obtain the speech quantization sequence.

[0071] Specifically, based on the codebook in the VQ-VAE model, the continuous speech encoding vector is quantized into a discrete speech quantization sequence through the nearest neighbor search algorithm.

[0072] In an embodiment, FIG. **4** shows a schematic flowchart of a method for performing quantization to obtain a speech quantization sequence. As shown in FIG. **4**, the method includes at least steps S**410** and S**420**. In the step S**410**, the codebook is updated to obtain an updated codebook.

[0073] In an embodiment, FIG. **5** shows a schematic flowchart of a method for updating a codebook. As shown in FIG. **5**, the method includes at least steps S**510** and S**520**. In the step S**510**, a codebook identity of each frame of codebook is obtained, and codebook identities are compared to obtain a comparison result.

[0074] Each frame of speech feature vector corresponds to a codebook, and for example, codebook representations of five frames of speech feature vectors may be book 1, book 1, book 1, book 2 and book 2.

[0075] Considering that the text feature vector may be subsequently processed in a sequence-to-sequence model, the codebook identities may be compared to obtain the comparison result, in order to obtain a sequence representation that better matches the text feature vector.

[0076] The comparison result may reflect whether the two or more codebook identities are the same.

[0077] In the step S**520**, codebooks are merged according to the comparison result to obtain the updated codebook.

[0078] When the comparison result shows that the two or more codebook identities are the same, the same codebook identities may be merged into one for query by the nearest neighbor search algorithm.

[0079] For example, when codebooks of five frames of speech feature vectors are represented as book 1, book 1, book 1, book 2 and book 2, the codebook identities may be merged into book 1 and book 2 to function as the updated codebook.

[0080] In the embodiments, updating the codebook through the post-processing network enables the speech quantization sequence to have a sequence representation that better matches the text feature vector, providing data support for a multimodal voice conversion task and facilitating the improvement of the voice conversion performance.

[0081] In the step S**420**, based on the updated codebook, the quantization is performed on the speech encoding vector through the nearest neighbor search algorithm to obtain the speech quantization sequence.

[0082] Each updated codebook is a K*D dimensional codebook maintained in the VQ-VAE model.

[0083] For example, each codebook may include K D-dimensional encoding vectors e.sub.1, e.sub.2, . . . , e.sub.K. The encoding layer of the VQ-VAE model is used to encode the H′*W′*D dimensional speech feature vector, and then for each D-dimensional vector in the H′*W′*D dimensional speech feature vector, an encoding vector e.sub.i closest to this D-dimensional vector may be found in the codebook, respectively. The encoding vector e.sub.i is a vector in the codebook, and the D-dimensional vector is represented by an index of the encoding vector e.sub.i, obtaining the discrete H′*W′ dimensional vector, where K, D, H′ and W′ represent dimensions, respectively.

[0084] Furthermore, according to a preset discrete encoding method, the discrete H′*W′ dimensional vector is converted into the speech quantization sequence.

[0085] The preset discrete encoding method may be one-hot encoding or other types of encoding methods, which is not particularly limited in the embodiments.

[0086] Specifically, a codebook with the one-hot encoding method is used to convert, by means of the lookup table, the discrete H′*W′ dimensional vector into another discrete H′*W′ dimensional coding vector which is encoded by the codebook with the one-hot encoding method, and then the speech quantization sequence is obtained according to the converted discrete H′*W′ dimensional encoding vector.

[0087] For example, after a discrete 3*3 vector is converted into another discrete 3*3 encoding vector encoded in a codebook with the one-hot coding method, a 1*9 speech quantization sequence may be obtained according to each element in the converted discrete 3*3 encoding vector.

[0088] In the embodiments, the discrete encoding processing is performed by the vector quantization module of the speech generation model on the speech encoding vector to obtain the corresponding speech quantization sequence, which provides a data basis and theoretical support for the speech generation model to output the speech encoding vector and the self-restored speech.

[0089] In addition, a speaker vector of a speaker who speaks or emits the to-be-processed speech may also be obtained.

[0090] In an embodiment, FIG. **6** shows a schematic flowchart of a method for acquiring a speaker vector. As shown in FIG. **6**, the method includes at least steps S**610** and S**620**. In the step S**610**, a speaker identity corresponding to the to-be-processed speech is acquired, and a correspondence between the speaker identity and the speaker vector is determined, the correspondence is determined according to the speech generation model.

[0091] The speaker identity may uniquely represent identification information of the speaker who speaks or emits the to-be-processed speech.

[0092] It should be noted that a table storing the correspondence between the speaker identity and the speaker vector can be also maintained through a first loss value calculated between the self-restored speech and the to-be-processed speech which is output by the decoder in the speech generation model.

[0093] In the step S**620**, the speaker vector corresponding to the speaker identity is queried according to the correspondence.

[0094] In the table storing the correspondence between the speaker identity and the speaker vector, the corresponding speaker vector may be queried according to the speaker identity.

[0095] In the embodiments, by acquiring the speaker vector maintained by the speech generation model, data support is provided for the decoder module of the speech generation model, and the support provided by the training of the decoder module and the encoder module through the speaker vector and the speech quantization sequence can help the generation by the encoder of the speech generation model and the determination of the speaker vector.

[0096] In the step S**330**, the nonlinear transformation is performed, by using a decoder module of the speech generation model, on the speech quantization sequence and the speaker vector to obtain the self-restored speech.

[0097] After receiving the quantized speech quantization sequence and acquiring the speaker vector, the decoder module of the speech generation model may sum the speech quantization sequence and the speaker vector, and then restore the self-restored speech through the nonlinear transformation.

[0098] The decoder converts the high-dimensional and abstract hidden feature into the more explicit feature through the nonlinear transformation of the neural network.

[0099] It should be noted that when the speech generation model is the VQ-VAE model, the decoder module in the VQ-VAE model may also be composed of CNN and LSTM.

[0100] In the embodiments, through the corresponding processing of the encoder module, the vector quantization module and the decoder module in the speech generation model, it is possible to output the self-restored speech to support the training of the speech generation model.

[0101] In the step S**220**, loss calculation is performed on the to-be-processed speech and the self-restored speech to obtain a first loss value, and the speech encoding vector is determined as the language unit vector according to the first loss value.

[0102] After the speech generation model outputs the self-restored speech, the first loss value of the speech generation model may be calculated through the to-be-processed speech and the self-restored speech.

[0103] Specifically, the first loss value may be calculated by means of an L2 norm loss function.

[0104] The L2 norm loss function is shown in formula (1):

[00001] $L = \text{.Math.}_{i=1}^{n} (y_i - f(x_i))^2 \quad (1)$

[0105] The L2 norm loss function is also called the Least Square Error (LSE). The L2 norm loss function minimizes the sum of the squares of the difference between the target value y.sub.i and the estimated value f(x.sub.i).

[0106] This loss is generally used in regression problems, and outliers have a greater impact on this loss.

[0107] When the first loss value calculated according to formula (1) reaches a stable value and no longer decreases, it indicates that the speech generation model has been trained.

[0108] In this case, the speech encoding vector output by the encoder module in the speech generation model may be determined as an unsupervised language unit vector. The language unit vector may be represented by k.sub.i.

[0109] In the embodiments, the language unit vector may be obtained through the speech generation model, which provides the sequence-to-sequence model with a data input in a speech modality and provides support for a multi-module speech generation method.

[0110] In the step S**120**, the text feature vector is acquired, and the to-be-processed feature vector is determined according to the text feature vector and the language unit vector.

[0111] In an embodiment of the present disclosure, the TTS is a system that automatically converts a natural text into the speech.

[0112] Therefore, after the natural text is acquired, a phoneme sequence of the natural text may be extracted as the text feature vector. The phoneme sequence of the natural text may be extracted by using an LSTM model, which is not particularly limited in the embodiments.

[0113] After the text feature vector is acquired, the to-be-processed feature vector may be determined according to the text feature vector and the language unit vector.

[0114] In an embodiment, FIG. **7** shows a schematic flowchart of a method for determining a to-be-processed feature vector according to a text feature vector and a language unit vector. As shown in FIG. **7**, the method includes at least steps S**710** and S**720**. In the step S**710**, the text feature vector or the language unit vector is determined as the to-be-processed feature vector.

[0115] In the TTS field, the speaker adaptation can be divided into supervised speaker adaptation and unsupervised speaker adaptation. The supervised speaker adaptation means that <text, speech> paired data is required during adaptation, while the unsupervised speaker adaptation means that only speech data is required during the adaptation without the corresponding text.

[0116] Previous researches have shown that in the supervised speaker adaptation, a small amount of <text, speech> paired data of the target speaker is used to fine-tune a multi-speaker base model, thereby achieving a high-quality effect.

[0117] However, in the unsupervised speaker adaptation, fine-tuning the model is not possible.

[0118] A common unsupervised speaker adaptation method uses a speaker recognition model to extract a speaker vector from a speech, and then uses this speaker vector to synthesize the speech of the speaker.

[0119] It is obvious that the unsupervised speaker adaptation is not achieved by the trained model, and the performance of this method will not improve much as the amount of data increases.

[0120] Therefore, in the unsupervised speaker adaptation, the text feature vector may be determined as the to-be-processed feature vector to achieve the TTS effect through the subsequent sequence-to-sequence model and vocoder.

[0121] For the voice conversion, the language unit vector may be determined as the to-be-processed feature vector so as to complete the voice conversion task through the subsequent sequence-to-sequence model and vocoder.

[0122] In the step S**720**, the text feature vector and the language unit vector are summed to obtain the to-be-processed feature vector.

[0123] In order to improve the effect of the voice conversion task, the text feature vector and the language unit vector may be summed to obtain the to-be-processed feature vector.

[0124] It should be noted that since the codebook in the vector quantization module of the speech generation model has been updated in the process of obtaining the language unit vector, the language unit vector quite matches the text feature vector, and the text feature vector and the language unit vector may be directly summed.

[0125] When the to-be-processed feature vector is obtained by summing the text feature vector and the language unit vector, the to-be-processed feature vector is equivalent to adding the performance of the text modality on the basis of the language modality.

[0126] Therefore, the to-be-processed feature vector in this case is the enhanced data performance. Based on this, the performance of the voice conversion task implemented by the to-be-processed feature vector is better.

[0127] Modality refers to a source or form of information.

[0128] For example, a piece of information may be expressed in a plurality of forms such as speech, video, text, and images. Each form of expression of the information may be called a modality of the information. On this basis, multi-modality is the integration of text, speech, vision, action, environment and the like. MultiModal Machine Learning (MMML) refers to an ability to process and understand multi-source modal information through a machine learning method. For example, the current popular research direction is multimodal learning between images, videos, audio, and semantics.

[0129] In the embodiments, different to-be-processed feature vectors can be determined as the basis for the subsequent model processing, depending on the TTS task and the voice conversion task, providing data support for improving the performance of the TTS task and the voice conversion task.

[0130] In the step S**130**, the to-be-processed feature vector is input into the sequence-to-sequence model to obtain the acoustic feature vector, and the acoustic feature vector is input into the vocoder to obtain the target speech corresponding to the to-be-processed speech or the text feature vector.

[0131] In an embodiment of the present disclosure, after the to-be-processed feature vector is determined, the to-be-processed feature vector may be input into the sequence-to-sequence model to obtain the corresponding acoustic feature vector.

[0132] In an embodiment, FIG. **8** shows a schematic flowchart of a method for outputting an acoustic feature vector by a sequence-to-sequence model. As shown in FIG. **8**, the method includes at least steps S**810** and S**820**. In the step S**810**, a to-be-processed acoustic vector of the to-be-processed feature vector is acquired, and the to-be-processed feature vector and the to-be-processed acoustic vector are inputted into the sequence-to-sequence model to make the sequence-to-sequence model output a processed acoustic vector.

[0133] The to-be-processed acoustic vector may also be a Mel-spectrum feature vector.

[0134] In an embodiment, FIG. **9** shows a schematic flowchart of a processing method by a sequence-to-sequence model. As shown in FIG. **9**, the method may include at least steps S**910** to S**930**. In the step S**910**, the to-be-processed feature vector and the to-be-processed acoustic vector are input into the sequence-to-sequence model, and nonlinear mapping is performed, by using an encoder module of the sequence-to-sequence model, on the to-be-processed feature vector and the to-be-processed acoustic vector to obtain a spatial encoding vector.

[0135] The sequence-to-sequence model may be a sequence-to-sequence (Seq2seq) model based on an attention mechanism, or may be other models, which is not particularly limited in the embodiments.

[0136] When the sequence-to-sequence model is the Seq2seq model based on the attention mechanism, the sequence-to-sequence model may include the encoder module, an attention mechanism, and a decoder module.

[0137] The encoder module of the sequence-to-sequence model may be configured to acquire a representation sequence corresponding to the to-be-processed feature vector and the to-be-processed acoustic vector, the attention mechanism may be configured to generate a fixed-length semantic representation based on the representation sequence, and the decoder module may be configured to acquire the acoustic vector based on the semantic representation.

[0138] Specifically, the encoder module of the sequence-to-sequence model may include a FeatureEmbedding layer, a Convolutional Pre-Net, a Dense Pre-Net, a Convolution Bank+Highway network+bidirectional Gated Recurrent Unit (CBHG, that is, the CBHG consists of the convolutional bank, the highway network, and the bidirectional gated recurrent unit) sub-model, a Down-sampling Convolution layer.

[0139] First, the to-be-processed feature vector is encoded using the FeatureEmbedding layer and then input into the Convolutional Pre-net to perform the nonlinear transformation on the encoded to-be-processed feature vector and the to-be-processed acoustic vector, thereby improving the convergence and generalization ability of the sequence-to-sequence model based on the attention mechanism. In addition, the number of speech frames corresponding to the to-be-processed acoustic vector is input into the Dense Pre-net to obtain the corresponding deep feature. Then, an output of the Convolutional Pre-net and an output of the Dense Pre-net are input into the CBHG sub-model together to extract the corresponding contextual feature, which is then input into the Down-sampling Convolution to reduce the amount of calculation and the receptive field, and the corresponding spatial encoding vector is finally obtained.

[0140] Therefore, the to-be-processed feature vector and the to-be-processed acoustic vector are nonlinearly transformed and mapped to a high-dimensional spatial encoding vector through the encoder module of the sequence-to-sequence model. The spatial encoding vector may be represented by h.sub.t.

[0141] In the step S**920**, the spatial encoding vector and the speaker vector are summed to obtain a to-be-aligned vector, and a speech feature sequence is acquired.

[0142] Since multi-speaker modeling is required, the attention mechanism of the sequence-to-sequence model may also receive the speaker vector as the input.

[0143] To input the speaker vector, the spatial encoding vector and the speaker vector may be summed to obtain the to-be-aligned vector.

[0144] Furthermore, since the attention mechanism is an autoregressive model, the speech feature sequence may also be acquired. The speech feature sequence may be represented by m.sub.t-1. When t=1, the speech feature sequence is initialized to a sequence of all 0 s; and when t=2 and subsequent times, the speech feature sequence is a feedback sequence of the decoder module for the previous moment.

[0145] In the step S**930**, the to-be-aligned vector and the speech feature sequence are aligned, by using the attention mechanism of the sequence-to-sequence model, to obtain a contextual representation vector, and the nonlinear mapping is performed, by using the decoder of the sequence-to-sequence model, on the contextual representation vector to obtain the processed acoustic vector.

[0146] Since the speech feature vector is usually longer than the to-be-aligned vector, the to-be-aligned vector and the speech feature sequence may be aligned to obtain the contextual representation vector.

[0147] Specifically, a manner to align the to-be-aligned vector and the speech feature sequence may be performing a dot product calculation on the to-be-aligned vector and the speech feature sequence.

[0148] Moreover, the contextual representation vector obtained by aligning the to-be-aligned vector and the speech feature sequence can reflect a contextual relationship of the context, thus ensuring the effect of speech generation.

[0149] Furthermore, the decoder module of the sequence-to-sequence model mainly returns, through the nonlinear mapping, the contextual representation vector obtained by aligning the to-be-aligned vector and the speech feature sequence to an original speech acoustic feature space, so as to obtain the processed acoustic vector. Therefore, the processed acoustic vector may be a Mel spectrum, and the processed acoustic vector may be represented by m.

[0150] In the embodiments, by correspondingly processing the to-be-processed feature vector and the to-be-processed acoustic vector by the encoder module, the attention mechanism and the decoder module in the sequence-to-sequence model, a fusion method can be provided for the TTS task and the VC task, thereby improving the voice cloning effect under a small amount of data. In addition, since a variety of input data can be received, the voice cloning in a variety of scenarios is supported.

[0151] In the step S**820**, the loss calculation is performed on the to-be-processed acoustic vector and the processed acoustic vector to obtain a second loss value, and the processed acoustic vector is determined as the acoustic feature vector according to the second loss value.

[0152] After the sequence-to-sequence model outputs the processed acoustic vector, the second loss value between the to-be-processed acoustic vector and the processed acoustic vector may be calculated according to the above formula (1).

[0153] When the second loss value calculated according to the formula (1) reaches a stable value and no longer decreases, it indicates that the sequence-to-sequence model has been trained.

[0154] In this case, the processed acoustic vector output by the sequence-to-sequence model which has been trained to be convergent may be determined as the acoustic feature vector.

[0155] After the sequence-to-sequence model outputs the acoustic feature vector, the acoustic feature vector may be further input into the vocoder to obtain the target speech for the TTS task or the voice conversion.

[0156] In an embodiment, FIG. **10** shows a schematic flowchart of a method for outputting a target speech according to a generator. As shown in FIG. **10**, the method includes at least steps S**1010** and S**1020**. In the step S**1010**, a speech acoustic feature of the acoustic feature vector is extracted by a post-processing network, and the speech acoustic feature is input into the vocoder to make the vocoder output a pending speech corresponding to the to-be-processed speech or the text feature vector.

[0157] The post-processing network is mainly set up to improve the generation of the higher-precision speech acoustic feature. The speech acoustic feature may be represented by {circumflex over (x)}.

[0158] The post-processing network may be a CNN network, or an LSTM network, etc., which is not particularly limited in the embodiments.

[0159] The vocoder is a system that converts the acoustic feature, such as the Mel spectrum, into a speech audio.

[0160] The vocoder may be a Wavenet model, a Griffin-Lim algorithm, a Generative Adversarial Network (GAN), or the like, which is not particularly limited in the embodiments.

[0161] Specifically, the Wavenet model is a sequence generation model that may be used for speech generation modeling. In the TTS acoustic model modeling, the Wavenet may directly learn the mapping of the sample value sequence, so it has a good synthesis effect.

[0162] At present, the Wavenet has applications in the TTS acoustic model modeling and the vocoder, and has great potential in the TTS field.

[0163] The Wavenet model may predict a result of the t-th point based on the first t−1 points of a sequence, so it may be used to predict a numerical value of the sampling point in the speech.

[0164] The Griffin-Lim is an algorithm for reconstructing a speech under a condition that only an amplitude spectrum is known but a phase spectrum is unknown.

[0165] The implementation for the Griffin-Lim algorithm is relatively simple. The Griffin-Lim algorithm is an iterative algorithm. The iterative process is to first randomly initialize a phase spectrum, then the phase spectrum and the known amplitude spectrum are used to synthesize a new speech through the inverse short-time Fourier transform (ISTFT), and then the short-time Fourier transform or the short-term Fourier transform (STFT) is performed on the synthesized speech to obtain a new amplitude spectrum and phase spectrum. Finally, the new amplitude spectrum is discarded, the phase spectrum and the known amplitude spectrum are used to synthesize the speech. The repetition is performed in this way.

[0166] The GAN network is a machine learning method proposed by Ian J. Goodfello et al. in the paper of Generative Adversarial Nets in 2014. In the GAN network, there are two models, which are respectively a generative model G and a discriminative model D.

[0167] Taking the picture generation as an example, G is a network that generates the picture, and it accepts a random noise z and then generates the picture through this noise.

[0168] The generated data is recorded as G(z).

[0169] D is a discriminative network that discriminates whether a picture is “real” (i.e., whether it is fabricated). Its input parameter is x, which represents a picture, and its output D(x) represents a probability of x being a real picture. If the output is 1, it means it is a real picture, and if the output is 0, it means it cannot be the real picture.

[0170] During the training process, a goal of the generative network G is to generate a fake image to deceive the discriminative network D, while a goal of the discriminative network D is to be able to discriminate whether a certain picture is generated by G. This becomes a game process. In addition, the capabilities of G and D are gradually improved during the training process. In the most ideal case, D (G(z))=0.5.

[0171] The speech acoustic feature extracted by the post-processing network is processed by the vocoder to output the pending speech.

[0172] When the TTS task is performed, the pending speech may be a speech synthesized according to the text feature vector, and when the VC task is performed, the pending speech may be a speech converted according to the to-be-processed speech.

[0173] In the step S**1020**, the loss calculation is performed on the pending speech and the to-be-processed speech to obtain a third loss value, and the pending speech is determined as the target speech according to the third loss value.

[0174] In an embodiment, FIG. **11** shows a schematic flowchart of a method for performing loss calculation to obtain a third loss value. As shown in FIG. **11**, the method includes at least steps S**1110** and S**1120**. In the step S**1110**, if the vocoder is a generative adversarial network, the loss calculation is performed on the pending speech and the to-be-processed speech to obtain an adversarial network loss value of the generative adversarial network.

[0175] When the vocoder adopts the generative adversarial network, a loss function of the generative adversarial network is shown in formula (2):

[00002] $\min_{G}\max_{D} V(D,G) = E_{x\sim P_{\text{data}}(x)}[\log(D(x))] + E_{z\sim P_z(z)}[\log(1 - D(G(z)))]$ (2)

[0176] where, D(x) represents discrimination for a real sample. Here, it is desired that its discrimination result is as close to 1 as possible, so the loss function is log(D(x)). z is a random input, and G(z) represents the generated sample. For the generated sample, it is desired that the discrimination result D(G(z)) of the discriminator is as close to 0 as possible, that is, to maximize the total value, so the overall expression is shown in formula (2).

[0177] Therefore, by performing the loss calculation on the pending speech and the to-be-processed speech according to the formula (2), the adversarial network loss value of the generative adversarial network can be obtained.

[0178] In the step S**1120**, the loss calculation is performed on the pending speech and the to-be-processed speech to obtain a speech feature loss value, and a weighted sum is performed on the adversarial network loss value and the speech feature loss value to obtain a third loss value.

[0179] Furthermore, the loss calculation is performed on the pending speech and the to-be-processed speech according to formula (1) to obtain the speech feature loss value.

[0180] Furthermore, corresponding weights may be set for the adversarial network loss value and the speech feature loss value according to empirical values, so as to perform the weighted sum calculation on the adversarial network loss value and the speech feature loss value to obtain the third loss value.

[0181] It should be noted that when the vocoder adopts other networks or models, the corresponding loss value may be calculated only according to formula (1) as the third loss value.

[0182] In the embodiments, the corresponding loss value calculation methods are set according to different vocoder contents, which is more targeted and can ensure the accuracy of training results of different types of vocoders, further ensuring the reliability of the target speech generation.

[0183] When the third loss value reaches a stable value and no longer decreases, it indicates that the vocoder has been trained to be convergent and can be put into an application stage. Therefore, it can be determined that the pending speech is the target speech.

[0184] The following is a detailed description of the method for generating the speech in embodiments of the present disclosure in conjunction with an application scenario.

[0185] FIG. **12** shows a schematic diagram of a framework of a speech generation model in an application scenario. As shown in FIG. **12**, the VQ-VAE model includes an encoder module **1210**, a vector quantization module **1220** and a decoder module **1230**.

[0186] First, a speech feature vector of a speech is input into the encoder module **1210** of the VQ-VAE model.

[0187] The to-be-processed speech may be a speech that needs to be converted for voice conversion. Correspondingly, the speech feature vector of the to-be-processed speech may be a Mel-spectrum feature vector extracted according to the to-be-processed speech

[0188] Furthermore, the nonlinear transformation is performed, by using the encoder module of the speech generation model, on the speech feature vector to obtain a high-dimensional speech encoding vector. The speech encoding vector may be represented by Z.sub.1:N.

[0189] The encoder module in the VQ-VAE model may be composed of CNN and LSTM.

[0190] Then, quantization is performed, by using the vector quantization module **1220** of the speech generation model, on the speech encoding vector to obtain a speech quantization sequence.

[0191] After the encoder module of the VQ-VAE model obtains the speech encoding vector Z.sub.1:N, the vector quantization ({circumflex over (z)}.sub.i=e.sub.k) module in the VQ-VAE model may be used to quantize the high-dimensional speech encoding vector into the speech quantization sequence. The speech quantization sequence may be represented by {circumflex over (z)}.sub.i:N.

[0192] Based on a codebook in the vector quantization module of the speech generation model, the quantization is performed on the speech encoding vector through a nearest neighbor search algorithm to obtain the speech quantization sequence.

[0193] Specifically, based on the updated codebook in the VQ-VAE model, the continuous speech encoding vector is quantized into a discrete speech quantization sequence through the nearest neighbor search algorithm.

[0194] When the codebook is updated, the codebook identity of each frame of codebook may be acquired, and the codebook identities may be compared to obtain the comparison result.

[0195] Each frame of speech feature vector corresponds to a codebook, and for example, codebook representations of five frames of speech feature vectors may be book 1, book 1, book 1, book 2 and book 2.

[0196] Considering that the text feature vector may be subsequently processed in a sequence-to-sequence model, the codebook identities may be compared to obtain the comparison result, in order to obtain a sequence representation that better matches the text feature vector.

[0197] The comparison result may reflect whether the two or more codebook identities are the same.

[0198] Codebooks are merged according to the comparison result to obtain the updated codebook.

[0199] When the comparison result shows that the two or more codebook identities are the same, the same codebook identities may be merged into one for query by the nearest neighbor search algorithm.

[0200] For example, when codebooks of five frames of speech feature vectors are represented as book 1, book 1, book 1, book 2 and book 2, the codebook identities may be merged into book 1 and book 2 to function as the updated codebook.

[0201] Based on the updated codebook, the quantization is performed on the speech encoding vector through the nearest neighbor search algorithm to obtain the speech quantization sequence.

[0202] Each updated codebook is a K*D dimensional codebook maintained in the VQ-VAE model.

[0203] For example, each codebook may include K D-dimensional encoding vectors e.sub.1, e.sub.2, . . . , e.sub.K. The encoding layer of the VQ-VAE model is used to encode the H′*W′*D dimensional speech feature vector, and then for each D-dimensional vector in the H′*W′*D dimensional speech feature vector, an encoding vector e.sub.1 closest to this D-dimensional vector may be found in the codebook, respectively. The encoding vector e.sub.i is a vector in the codebook, and the D-dimensional vector is represented by an index of the encoding vector et, obtaining the discrete H′*W′ dimensional vector, where K, D, H′ and W′ represent dimensions, respectively.

[0204] Furthermore, according to a preset discrete encoding method, the discrete H′*W′ dimensional vector is converted into the speech quantization sequence.

[0205] The preset discrete encoding method may be one-hot encoding or other types of encoding methods, which is not particularly limited in the embodiments.

[0206] Specifically, a codebook with the one-hot encoding method is used to convert, by means of the lookup table, the discrete H′*W′ dimensional vector into another discrete H′*W′ dimensional coding vector which is encoded by the codebook with the one-hot encoding method, and then the speech quantization sequence is obtained according to the converted discrete H′*W′ dimensional encoding vector.

[0207] For example, after a discrete 3*3 vector is converted into another discrete 3*3 encoding vector encoded in a codebook with the one-hot coding method, a 1*9 speech quantization sequence may be obtained according to each element in the converted discrete 3*3 encoding vector.

[0208] In addition, a speaker vector of a speaker who speaks or emits the to-be-processed speech may also be obtained.

[0209] A speaker identity corresponding to the to-be-processed speech is acquired, and a correspondence between the speaker identity and the speaker vector is determined, the correspondence is determined according to the speech generation model

[0210] The speaker identity may uniquely represent identification information of the speaker who speaks or emits the to-be-processed speech.

[0211] It should be noted that a table storing the correspondence between the speaker identity and the speaker vector can be also maintained through a first loss value calculated between the self-restored speech and the to-be-processed speech which is output by the decoder in the speech generation model.

[0212] The speaker vector corresponding to the speaker identity is queried according to the correspondence.

[0213] In the table storing the correspondence between the speaker identity and the speaker vector, the corresponding speaker vector may be queried according to the speaker identity.

[0214] Finally, the nonlinear transformation is performed, by using a decoder module of the speech generation model, on the speech quantization sequence and the speaker vector to obtain the self-restored speech.

[0215] After receiving the quantized speech quantization sequence and acquiring the speaker vector, the decoder module of the speech generation model may sum the speech quantization sequence and the speaker vector, and then restore the self-restored speech through the nonlinear transformation.

[0216] Loss calculation is performed on the to-be-processed speech and the self-restored speech to obtain a first loss value, and the speech encoding vector is determined as the language unit vector according to the first loss value.

[0217] After the speech generation model outputs the self-restored speech, the first loss value of the speech generation model may be calculated through the to-be-processed speech and the self-restored speech.

[0218] Specifically, the first loss value may be calculated by means of an L2 norm loss function. The L2 norm loss function is shown in the above formula (1).

[0219] When the first loss value calculated according to formula (1) reaches a stable value and no longer decreases, it indicates that the speech generation model has been trained.

[0220] In this case, the speech encoding vector output by the encoder module in the speech generation model may be determined as an unsupervised language unit vector. The language unit vector may be represented by k.sub.i.

[0221] The unsupervised learning can discover or extract the useful information representation through its own data. The VQ-VAE unsupervised algorithm in this application scenario can extract the discrete information representation from data in different formats.

[0222] This discrete representation unit is very close to the phoneme in the language text, and it is very appropriate to use this unsupervised discrete language unit as an input of an end-to-end language synthesis model.

[0223] Moreover, it is also a perfect fit for the problem to be solved.

[0224] In order to combine the TTS task and the VC task into one system, a common input of this system can be looked for, that is, the phoneme extracted from the text and the unsupervised language unit extracted by the VQ-VAE model.

[0225] In FIG. **12**, a sequence-to-sequence model may include an encoder module **1240**, an attention mechanism **1250**, a decoder module **1260**, and a post-processing network **1270**.

[0226] In order to the fusion of the TTS task and the VC task, the text feature vector may also be acquired.

[0227] After a natural text is acquired, a phoneme sequence of the natural text may be extracted as the text feature vector.

[0228] After the text feature vector is acquired, a to-be-processed feature vector may be determined according to the text feature vector and the language unit vector.

[0229] In the unsupervised speaker adaptation, the text feature vector may be determined as the to-be-processed feature vector to achieve the TTS effect through the subsequent sequence-to-sequence model and vocoder.

[0230] For the voice conversion, the language unit vector may be determined as the to-be-processed feature vector so as to complete the voice conversion task through the subsequent sequence-to-sequence model and vocoder.

[0231] In order to improve the effect of the voice conversion task, the text feature vector and the language unit vector may be summed to obtain the to-be-processed feature vector.

[0232] It should be noted that since the codebook in the vector quantization module of the speech generation model has been updated in the process of obtaining the language unit vector, the language unit vector quite matches the text feature vector, and the text feature vector and the language unit vector may be directly summed.

[0233] When the to-be-processed feature vector is obtained by summing the text feature vector and the language unit vector, the to-be-processed feature vector is equivalent to adding the performance of the text modality on the basis of the language modality.

[0234] Therefore, the to-be-processed feature vector in this case is the enhanced data performance. Based on this, the performance of the voice conversion task implemented by the to-be-processed feature vector is better.

[0235] Furthermore, the to-be-processed acoustic vector of the to-be-processed feature vector is acquired, and the to-be-processed acoustic vector may also be a Mel-spectrum feature vector.

[0236] The to-be-processed feature vector and the to-be-processed acoustic vector are input into the sequence-to-sequence model, and nonlinear mapping is performed, by using the encoder module **1240** of the sequence-to-sequence model, on the to-be-processed feature vector and the to-be-processed acoustic vector to obtain a spatial encoding vector.

[0237] The sequence-to-sequence model may be a sequence-to-sequence model based on an attention mechanism.

[0238] Specifically, the encoder module of the sequence-to-sequence model may include a FeatureEmbedding layer, a Convolutional Pre-Net, a Dense Pre-Net, a CBHG sub-model, a Down-sampling Convolution layer.

[0239] First, the to-be-processed feature vector is encoded using the FeatureEmbedding layer and then input into the Convolutional Pre-net to perform the nonlinear transformation on the encoded to-be-processed feature vector and the to-be-processed acoustic vector, thereby improving the convergence and generalization ability of the sequence-to-sequence model based on the attention mechanism. In addition, the number of speech frames corresponding to the to-be-processed acoustic vector is input into the Dense Pre-net to obtain the corresponding deep feature. Then, an output of the Convolutional Pre-net and an output of the Dense Pre-net are input into the CBHG sub-model together to extract the corresponding contextual feature, which is then input into the Down-sampling Convolution to reduce the amount of calculation and the receptive field, and the corresponding spatial encoding vector is finally obtained.

[0240] Therefore, the to-be-processed feature vector and the to-be-processed acoustic vector are nonlinearly transformed and mapped to a high-dimensional spatial encoding vector through the encoder module of the sequence-to-sequence model. The spatial encoding vector may be represented by h.sub.t.

[0241] The spatial encoding vector and the speaker vector are summed to obtain a to-be-aligned vector, and a speech feature sequence is acquired.

[0242] Since multi-speaker modeling is required, the attention mechanism **1250** of the sequence-to-sequence model may also receive the speaker vector as the input.

[0243] To input the speaker vector, the spatial encoding vector and the speaker vector may be summed to obtain the to-be-aligned vector.

[0244] Furthermore, since the attention mechanism **1250** is an autoregressive model, the speech feature sequence may also be acquired. The speech feature sequence may be represented by m.sub.t-1. When t=1, the speech feature sequence is initialized to a sequence of all 0 s; and when t=2 and subsequent times, the speech feature sequence is a feedback sequence of the decoder module **1260** for the previous moment.

[0245] The to-be-aligned vector and the speech feature sequence are aligned, by using the attention mechanism **1250** of the sequence-to-sequence model, to obtain a contextual representation vector, and the nonlinear mapping is performed, by using the decoder **1260** of the sequence-to-sequence model, on the contextual representation vector to obtain the processed acoustic vector.

[0246] Since the speech feature vector is usually longer than the to-be-aligned vector, the to-be-aligned vector and the speech feature sequence may be aligned to obtain the contextual representation vector.

[0247] Specifically, a manner to align the to-be-aligned vector and the speech feature sequence may be performing a dot product calculation on the to-be-aligned vector and the speech feature sequence.

[0248] Moreover, the contextual representation vector obtained by aligning the to-be-aligned vector and the speech feature sequence can reflect a contextual relationship of the context, thus ensuring the effect of speech generation.

[0249] Furthermore, the decoder module **1260** of the sequence-to-sequence model mainly returns, through the nonlinear mapping, the contextual representation vector obtained by aligning the to-be-aligned vector and the speech feature sequence to an original speech acoustic feature space, so as to obtain the processed acoustic vector. Therefore, the processed acoustic vector may be a Mel spectrum, and the processed acoustic vector may be represented by m.

[0250] After the sequence-to-sequence model outputs the processed acoustic vector, the second loss value between the to-be-processed acoustic vector and the processed acoustic vector may be calculated according to the above formula (1).

[0251] When the second loss value calculated according to the formula (1) reaches a stable value and no longer decreases, it indicates that the sequence-to-sequence model has been trained.

[0252] In this case, the processed acoustic vector output by the sequence-to-sequence model which has been trained to be convergent may be determined as the acoustic feature vector.

[0253] A speech acoustic feature of the acoustic feature vector is extracted by the post-processing network **1270**, and the speech acoustic feature is input into the vocoder **1280** to make the vocoder **1280** output a pending speech corresponding to the to-be-processed speech or the text feature vector.

[0254] The post-processing network **1270** is mainly set up to improve the generation of the higher-precision speech acoustic feature. The speech acoustic feature may be represented by {circumflex over (x)}.

[0255] The vocoder may be a Wavenet model, a Griffin-Lim algorithm, a GAN network, or the like, which is not particularly limited in the embodiments.

[0256] The speech acoustic feature extracted by the post-processing network **1270** is processed by the vocoder **1280** to output the pending speech.

[0257] When the TTS task is performed, the pending speech may be a speech synthesized according to the text feature vector, and when the VC task is performed, the pending speech may be a speech converted according to the to-be-processed speech.

[0258] If the vocoder **1280** is a generative adversarial network, the loss calculation is performed on the pending speech and the to-be-processed speech to obtain an adversarial network loss value of the generative adversarial network.

[0259] When the vocoder **1280** adopts the generative adversarial network, a loss function of the generative adversarial network is shown in the above formula (2). Therefore, by performing the loss calculation on the pending speech and the to-be-processed speech according to the formula (2), the adversarial network loss value of the generative adversarial network can be obtained.

[0260] The loss calculation is performed on the pending speech and the to-be-processed speech to obtain a speech feature loss value, and a weighted sum is performed on the adversarial network loss value and the speech feature loss value to obtain a third loss value.

[0261] Furthermore, the loss calculation is performed on the pending speech and the to-be-processed speech according to formula (1) to obtain the speech feature loss value.

[0262] Furthermore, corresponding weights may be set for the adversarial network loss value and the speech feature loss value according to empirical values, so as to perform the weighted sum calculation on the adversarial network loss value and the speech feature loss value to obtain the third loss value.

[0263] It should be noted that when the vocoder **1280** adopts other networks or models, the corresponding loss value may be calculated only according to formula (1) as the third loss value.

[0264] When the third loss value reaches a stable value and no longer decreases, it indicates that the vocoder **1280** has been trained to be convergent and can be put into an application stage. Therefore, it can be determined that the pending speech is the target speech.

[0265] In the method for generating the speech in embodiments of the present disclosure, by acquiring the speech feature vector and the text feature vector, the speech and the text can be received as the inputs, which facilitate the fusion of a TTS task and a VC task to perform multimodal modeling, and improve the performance of the TTS task and the VC task. Furthermore, the speech feature vector and the text feature vector are acquired in the case of a small amount of data, which provides a variety of voice cloning strategies, improves the effect of voice cloning under the small amount of data, reduces the training difficulty and training duration of various models, and supports voice cloning methods in various application scenarios.

[0266] In addition, the method for generating the speech in this application scenario can achieve a good effect in the supervised speaker adaptation through the fine-tuning, and in the unsupervised speaker adaptation, the performance can also be improved as the amount of data increases.

[0267] In the supervised voice cloning, this performance improvement is attributed to the use of the unsupervised language unit as a means of data augmentation, which can help the model perform better with fewer data samples.

[0268] In addition, whether in the supervised or the unsupervised voice cloning, the use of the unsupervised language unit can help the attention mechanism learn the more robust alignment result, thereby improving the performance of the model with few samples.

[0269] In the VC task, the performance of this method is also better than that of a single-task VC model, and the improvement is even greater.

[0270] Therefore, overall, the multimodal voice cloning system proposed in this application scenario can outperform the single-task TTS or VC model in many scenarios and has strong practical application value.

[0271] In addition, in embodiments of the present disclosure, there is further provided a device for generating a speech. FIG. **13** shows a schematic structural diagram of a device for generating a speech. As shown in FIG. **13**, the device **1300** for generating the speech may include a data acquisition module **1310**, a vector determination module **1320** and a speech generation module **1330**.

[0272] The data acquisition module **1310** is configured to acquire a speech feature vector of a to-be-processed speech, and input the speech feature vector into a speech generation model to obtain a language unit vector.

[0273] The vector determination module **1320** is configured to acquire a text feature vector, and determine a to-be-processed feature vector according to the text feature vector and the language unit vector.

[0274] The speech generation module **1330** is configured to input the to-be-processed feature vector into a sequence-to-sequence model to obtain an acoustic feature vector, and input the acoustic feature vector into a vocoder to obtain a target speech corresponding to the to-be-processed speech or the text feature vector.

[0275] In an embodiment of the present disclosure, inputting the speech feature vector into the speech generation model to obtain the language unit vector includes: [0276] inputting the speech feature vector into the speech generation model to make the speech generation model output a speech encoding vector and a self-restored speech; and [0277] performing loss calculation on the to-be-processed speech and the self-restored speech to obtain a first loss value, and determining the speech encoding vector as the language unit vector according to the first loss value.

[0278] In an embodiment of the present disclosure, inputting the speech feature vector into the speech generation model to make the speech generation model output the speech encoding vector and the self-restored speech includes: [0279] inputting the speech feature vector into the speech generation model, and performing, by using an encoder module of the speech generation model, nonlinear transformation on the speech feature vector to obtain the speech encoding vector; [0280] performing, by using a vector quantization module of the speech generation model, quantization on the speech encoding vector to obtain a speech quantization sequence, and acquiring a speaker vector corresponding to the to-be-processed speech; and [0281] performing, by using a decoder module of the speech generation model, the nonlinear transformation on the speech quantization sequence and the speaker vector to obtain the self-restored speech.

[0282] In an embodiment of the present disclosure, acquiring the speaker vector corresponding to the to-be-processed speech includes: [0283] acquiring a speaker identity corresponding to the to-be-processed speech, and determining a correspondence between the speaker identity and the speaker vector, wherein the correspondence is determined according to the speech generation model; and [0284] querying, according to the correspondence, the speaker vector corresponding to the speaker identity.

[0285] In an embodiment of the present disclosure, performing, by using the vector quantization module of the speech generation model, the quantization on the speech encoding vector to obtain the speech quantization sequence includes: [0286] based on a codebook in the vector quantization module of the speech generation model, performing the quantization on the speech encoding vector through a nearest neighbor search algorithm to obtain the speech quantization sequence.

[0287] In an embodiment of the present disclosure, performing the quantization on the speech encoding vector through the nearest neighbor search algorithm to obtain the speech quantization sequence includes: [0288] updating the codebook to obtain an updated codebook; and [0289] based on the updated codebook, performing the quantization on the speech encoding vector through the nearest neighbor search algorithm to obtain the speech quantization sequence.

[0290] In an embodiment of the present disclosure, updating the codebook to obtain the updated codebook includes: [0291] acquiring a codebook identity of each frame of codebook, and comparing codebook identities to obtain a comparison result; and [0292] merging codebooks according to the comparison result to obtain the updated codebook.

[0293] In an embodiment of the present disclosure, inputting the to-be-processed feature vector into the sequence-to-sequence model to obtain the acoustic feature vector includes: [0294] acquiring a to-be-processed acoustic vector of the to-be-processed feature vector, and inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model to make the sequence-to-sequence model output a processed acoustic vector; and [0295] performing the loss calculation on the to-be-processed acoustic vector and the processed acoustic vector to obtain a second loss value, and determining the processed acoustic vector as the acoustic feature vector according to the second loss value.

[0296] In an embodiment of the present disclosure, inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model to make the sequence-to-sequence model output the processed acoustic vector includes: [0297] inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model, and performing, by using an encoder module of the sequence-to-sequence model, nonlinear mapping on the to-be-processed feature vector and the to-be-processed acoustic vector to obtain a spatial encoding vector; [0298] summing the spatial encoding vector and the speaker vector to obtain a to-be-aligned vector, and acquiring a speech feature sequence; and [0299] aligning, by using an attention mechanism of the sequence-to-sequence model, the to-be-aligned vector and the speech feature sequence to obtain a contextual representation vector, and performing, by using a decoder of the sequence-to-sequence model, the nonlinear mapping on the contextual representation vector to obtain the processed acoustic vector.

[0300] In an embodiment of the present disclosure, determining the to-be-processed feature vector according to the text feature vector and the language unit vector includes: [0301] determining the text feature vector or the language unit vector as the to-be-processed feature vector; or [0302] summing the text feature vector and the language unit vector to obtain the to-be-processed feature vector.

[0303] In an embodiment of the present disclosure, inputting the acoustic feature vector into the vocoder to obtain the target speech corresponding to the to-be-processed speech or the text feature vector includes: [0304] extracting, by a post-processing network, a speech acoustic feature of the acoustic feature vector, and inputting the speech acoustic feature into the vocoder to make the vocoder output a pending speech corresponding to the to-be-processed speech or the text feature vector; and [0305] performing loss calculation on the pending speech and the to-be-processed speech to obtain a third loss value, and determining the pending speech as the target speech according to the third loss value.

[0306] In an embodiment of the present disclosure, performing the loss calculation on the pending speech and the to-be-processed speech to obtain the third loss value includes: [0307] if the vocoder is a generative adversarial network, performing the loss calculation on the pending speech and the to-be-processed speech to obtain an adversarial network loss value of the generative adversarial network; and [0308] performing the loss calculation on the pending speech and the to-be-processed speech to obtain a speech feature loss value, and performing a weighted sum on the adversarial network loss value and the speech feature loss value to obtain the third loss value.

[0309] Specific details of the device for generating the speech have been described in detail in the corresponding method for generating the speech, which will not be repeated here.

[0310] It should be noted that although several modules or units of the device for generating the speech are mentioned in the above detailed description, such division of modules or units is not mandatory. In fact, features and functions of two or more of the modules or units described above may be embodied in one module or unit in accordance with embodiments of the present disclosure. Conversely, the features and functions of one module or unit described above may be further divided into a plurality of modules or units.

[0311] In addition, in embodiments of the present disclosure, there is further provided an electronic device capable of implementing the above method.

[0312] An electronic device **1400** according to such embodiment of the present disclosure is described below with reference to FIG. **14**. The electronic device **1400** shown in FIG. **14** is only an example and should not bring any limitations to functions and a scope of use of embodiments of the present disclosure.

[0313] As shown in FIG. **14**, the electronic device **1400** takes the form of a general-purpose computing device. Components of the electronic device **1400** may include, but are not limited to: at least one processing unit **1410**, at least one storage unit **1420**, a bus **1430** connecting different system components (including the storage unit **1420** and the processing unit **1410**), and a display unit **1440**.

[0314] The storage unit stores program codes, and the program codes may be executed by the processing unit **1410**, so that the processing unit **1410** performs steps according to various embodiments of the present disclosure described in the above-mentioned "Example Method" section of this specification. For example, [0315] the method includes: [0316] acquiring a speech feature vector of a to-be-processed speech, and inputting the speech feature vector into a speech generation model to obtain a language unit vector; [0317] acquiring a text feature vector, and determining a to-be-processed feature vector according to the text feature vector and the language unit vector; and [0318] inputting the to-be-processed feature vector into a sequence-to-sequence model to obtain an acoustic feature vector, and inputting the acoustic feature vector into a vocoder to obtain a target speech corresponding to the to-be-processed speech or the text feature vector.

[0319] In an embodiment of the present disclosure, inputting the speech feature vector into the speech generation model to obtain the language unit vector includes: [0320] inputting the speech feature vector into the speech generation model to make the speech generation model output a speech encoding vector and a self-restored speech; and [0321] performing loss calculation on the to-be-processed speech and the self-restored speech to obtain a first loss value, and determining the speech encoding vector as the language unit vector according to the first loss value.

[0322] In an embodiment of the present disclosure, inputting the speech feature vector into the speech generation model to make the speech generation model output the speech encoding vector and the self-restored speech includes: [0323] inputting the speech feature vector into the speech generation model, and performing, by using an encoder module of the speech generation model, nonlinear transformation on the speech feature vector to obtain the speech encoding vector; [0324] performing, by using a vector quantization module of the speech generation model, quantization on the speech encoding vector to obtain a speech quantization sequence, and acquiring a speaker vector corresponding to the to-be-processed speech; and [0325] performing, by using a decoder module of the speech generation model, the nonlinear transformation on the speech quantization sequence and the speaker vector to obtain the self-restored speech.

[0326] In an embodiment of the present disclosure, acquiring the speaker vector corresponding to the to-be-processed speech includes: [0327] acquiring a speaker identity corresponding to the to-be-processed speech, and determining a correspondence between the speaker identity and the speaker vector, wherein the correspondence is determined according to the speech generation model; and

[0328] querying, according to the correspondence, the speaker vector corresponding to the speaker identity.

[0329] In an embodiment of the present disclosure, performing, by using the vector quantization module of the speech generation model, the quantization on the speech encoding vector to obtain the speech quantization sequence includes: [0330] based on a codebook in the vector quantization module of the speech generation model, performing the quantization on the speech encoding vector through a nearest neighbor search algorithm to obtain the speech quantization sequence.

[0331] In an embodiment of the present disclosure, performing the quantization on the speech encoding vector through the nearest neighbor search algorithm to obtain the speech quantization sequence includes: [0332] updating the codebook to obtain an updated codebook; and [0333] based on the updated codebook, performing the quantization on the speech encoding vector through the nearest neighbor search algorithm to obtain the speech quantization sequence.

[0334] In an embodiment of the present disclosure, updating the codebook to obtain the updated codebook includes: [0335] acquiring a codebook identity of each frame of codebook, and comparing codebook identities to obtain a comparison result; and [0336] merging codebooks according to the comparison result to obtain the updated codebook.

[0337] In an embodiment of the present disclosure, inputting the to-be-processed feature vector into the sequence-to-sequence model to obtain the acoustic feature vector includes: [0338] acquiring a to-be-processed acoustic vector of the to-be-processed feature vector, and inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model to make the sequence-to-sequence model output a processed acoustic vector; and [0339] performing the loss calculation on the to-be-processed acoustic vector and the processed acoustic vector to obtain a second loss value, and determining the processed acoustic vector as the acoustic feature vector according to the second loss value.

[0340] In an embodiment of the present disclosure, inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model to make the sequence-to-sequence model output the processed acoustic vector includes: [0341] inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model, and performing, by using an encoder module of the sequence-to-sequence model, nonlinear mapping on the to-be-processed feature vector and the to-be-processed acoustic vector to obtain a spatial encoding vector; [0342] summing the spatial encoding vector and the speaker vector to obtain a to-be-aligned vector, and acquiring a speech feature sequence; and [0343] aligning, by using an attention mechanism of the sequence-to-sequence model, the to-be-aligned vector and the speech feature sequence to obtain a contextual representation vector, and performing, by using a decoder of the sequence-to-sequence model, the nonlinear mapping on the contextual representation vector to obtain the processed acoustic vector.

[0344] In an embodiment of the present disclosure, determining the to-be-processed feature vector according to the text feature vector and the language unit vector includes: [0345] determining the text feature vector or the language unit vector as the to-be-processed feature vector; or [0346] summing the text feature vector and the language unit vector to obtain the to-be-processed feature vector.

[0347] In an embodiment of the present disclosure, inputting the acoustic feature vector into the vocoder to obtain the target speech corresponding to the to-be-processed speech or the text feature vector includes: [0348] extracting, by a post-processing network, a speech acoustic feature of the acoustic feature vector, and inputting the speech acoustic feature into the vocoder to make the vocoder output a pending speech corresponding to the to-be-processed speech or the text feature vector; and [0349] performing loss calculation on the pending speech and the to-be-processed speech to obtain a third loss value, and determining the pending speech as the target speech according to the third loss value.

[0350] In an embodiment of the present disclosure, performing the loss calculation on the pending speech and the to-be-processed speech to obtain the third loss value includes: [0351] if the vocoder is a generative adversarial network, performing the loss calculation on the pending speech and the to-be-processed speech to obtain an adversarial network loss value of the generative adversarial network; and [0352] performing the loss calculation on the pending speech and the to-be-processed speech to obtain a speech feature loss value, and performing a weighted sum on the adversarial network loss value and the speech feature loss value to obtain the third loss value.

[0353] Through the above method, by acquiring the speech feature vector and the text feature vector, the speech and the text can be received as the inputs, which facilitate the fusion of a TTS task and a VC task to perform multimodal modeling, and improve the performance of the TTS task and the VC task. Furthermore, the speech feature vector and the text feature vector are acquired in the case of a small amount of data, which provides a variety of voice cloning strategies, improves the effect of voice cloning under the small amount of data, reduces the training difficulty and training duration of various models, and supports voice cloning methods in various application scenarios.

[0354] The storage unit **1420** may include a readable medium in the form of a volatile storage unit, such as a random access storage unit (RAM) **1421** and/or a cache storage unit **1422**, and may further include a read-only storage unit (ROM) **1423**.

[0355] The storage unit **1420** may further include a program/utility tool **1424** having a set (at least one) of program modules **1425**. Such program modules **1425** include, but are not limited to, an operating system, one or more application programs, other program modules, and program data. Each or some combination of these examples may include an implementation of a network environment.

[0356] The bus **1430** may be one or more of several types of bus structures, including a memory unit bus or a memory unit controller, a peripheral bus, a graphics acceleration port, a processing unit, or a local area bus using any bus structure in a variety of bus structures.

[0357] The electronic device **1400** may also communicate with one or more external devices **1600** (such as a keyboard, a pointing device, a Bluetooth device, etc.), and may also communicate with one or more devices that enable a user to interact with the electronic device **1400**, and/or may also communicate with any device (such as a router, a modem) that can enable the electronic device **1400** to interact with one or more other computing devices. Such communication may be performed through an input/output (I/O) interface **1450**. Moreover, the electronic device **1400** may also communicate with one or more networks (such as a local area network (LAN), a wide area network (WAN), and/or a public network, such as the Internet) through the network adapter **1460**. As shown in the figure, the network adapter **1460** communicates with other modules of the electronic device **1400** through the bus **1430**. It should be understood that although not shown in the figure, other hardware and/or software modules may be used in conjunction with the electronic device **1400**, including but not limited to: microcode, device drivers, redundant processing units, external disk drive arrays, RAID systems, tape drives and data backup storage systems.

[0358] Through the description of the foregoing embodiments, those skilled in the art can easily understand that the example embodiments described herein may be implemented by software, or by software in combination with necessary hardware. Therefore, the technical solutions according to the embodiments of the present disclosure may be embodied in the form of a software product, and the software product may be stored in a non-volatile storage medium (which may be a CD-ROM, a U disk, a mobile hard disk, etc.) or on a network. The software product may include instructions to cause a computing device (which may be a personal computer, a server, a terminal device, or a network device, etc.) to execute the method according to embodiments of the present disclosure.

[0359] In an embodiment of the present disclosure, there is further provided a computer-readable storage medium having a program product stored thereon which is capable of implementing the method described above in this specification. In some embodiments of the present disclosure, various aspects of the present disclosure may also be implemented in the form of a program product, which includes program codes. When the program product is run on a terminal device, the program code is used to cause the terminal device to perform steps according to various embodiments of the present disclosure described in the above-mentioned “Example Method” section of this specification. For example, [0360] the method includes: [0361] acquiring a speech feature vector of a to-be-processed speech, and inputting the speech feature vector into a speech generation model to obtain a language unit vector; [0362] acquiring a text feature vector, and determining a to-be-processed feature vector according to the text feature vector and the language unit vector; and [0363] inputting the to-be-processed feature vector into a sequence-to-sequence model to obtain an acoustic feature vector, and inputting the acoustic feature vector into a vocoder to obtain a target speech corresponding to the to-be-processed speech or the text feature vector.

[0364] In an embodiment of the present disclosure, inputting the speech feature vector into the speech generation model to obtain the language unit vector includes: [0365] inputting the speech feature vector into the speech generation model to make the speech generation model output a speech encoding vector and a self-restored speech; and [0366] performing loss calculation on the to-be-processed speech and the self-restored speech to obtain a first loss value, and determining the speech encoding vector as the language unit vector according to the first loss value.

[0367] In an embodiment of the present disclosure, inputting the speech feature vector into the speech generation model to make the speech generation model output the speech encoding vector and the self-restored speech includes: [0368] inputting the speech feature vector into the speech generation model, and performing, by using an encoder module of the speech generation model, nonlinear transformation on the speech feature vector to obtain the speech encoding vector; [0369] performing, by using a vector quantization module of the speech generation model, quantization on the speech encoding vector to obtain a speech quantization sequence, and acquiring a speaker vector corresponding to the to-be-processed speech; and [0370] performing, by using a decoder module of the speech generation model, the nonlinear transformation on the speech quantization sequence and the speaker vector to obtain the self-restored speech.

[0371] In an embodiment of the present disclosure, acquiring the speaker vector corresponding to the to-be-processed speech includes: [0372] acquiring a speaker identity corresponding to the to-be-processed speech, and determining a correspondence between the speaker identity and the speaker vector, wherein the correspondence is determined according to the speech generation model; and

[0373] querying, according to the correspondence, the speaker vector corresponding to the speaker identity.

[0374] In an embodiment of the present disclosure, performing, by using the vector quantization module of the speech generation model, the quantization on the speech encoding vector to obtain the speech quantization sequence includes: [0375] based on a codebook in the vector quantization module of the speech generation model, performing the quantization on the speech encoding vector through a nearest neighbor search algorithm to obtain the speech quantization sequence.

[0376] In an embodiment of the present disclosure, performing the quantization on the speech encoding vector through the nearest neighbor search algorithm to obtain the speech quantization sequence includes: [0377] updating the codebook to obtain an updated codebook; and [0378] based on the updated codebook, performing the quantization on the speech encoding vector through the nearest neighbor search algorithm to obtain the speech quantization sequence.

[0379] In an embodiment of the present disclosure, updating the codebook to obtain the updated codebook includes: [0380] acquiring a codebook identity of each frame of codebook, and comparing codebook identities to obtain a comparison result; and [0381] merging codebooks according to the comparison result to obtain the updated codebook.

[0382] In an embodiment of the present disclosure, inputting the to-be-processed feature vector into the sequence-to-sequence model to obtain the acoustic feature vector includes: [0383] acquiring a to-be-processed acoustic vector of the to-be-processed feature vector, and inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model to make the sequence-to-sequence model output a processed acoustic vector; and [0384] performing the loss calculation on the to-be-processed acoustic vector and the processed acoustic vector to obtain a second loss value, and determining the processed acoustic vector as the acoustic feature vector according to the second loss value.

[0385] In an embodiment of the present disclosure, inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model to make the sequence-to-sequence model output the processed acoustic vector includes: [0386] inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model, and performing, by using an encoder module of the sequence-to-sequence model, nonlinear mapping on the to-be-processed feature vector and the to-be-processed acoustic vector to obtain a spatial encoding vector; [0387] summing the spatial encoding vector and the speaker vector to obtain a to-be-aligned vector, and acquiring a speech feature sequence; and [0388] aligning, by using an attention mechanism of the sequence-to-sequence model, the to-be-aligned vector and the speech feature sequence to obtain a contextual representation vector, and performing, by using a decoder of the sequence-to-sequence model, the nonlinear mapping on the contextual representation vector to obtain the processed acoustic vector.

[0389] In an embodiment of the present disclosure, determining the to-be-processed feature vector according to the text feature vector and the language unit vector includes: [0390] determining the text feature vector or the language unit vector as the to-be-processed feature vector; or [0391] summing the text feature vector and the language unit vector to obtain the to-be-processed feature vector.

[0392] In an embodiment of the present disclosure, inputting the acoustic feature vector into the vocoder to obtain the target speech corresponding to the to-be-processed speech or the text feature vector includes: [0393] extracting, by a post-processing network, a speech acoustic feature of the acoustic feature vector, and inputting the speech acoustic feature into the vocoder to make the vocoder output a pending speech corresponding to the to-be-processed speech or the text feature vector; and [0394] performing loss calculation on the pending speech and the to-be-processed speech to obtain a third loss value, and determining the pending speech as the target speech according to the third loss value.

[0395] In an embodiment of the present disclosure, performing the loss calculation on the pending speech and the to-be-processed speech to obtain the third loss value includes: [0396] if the vocoder is a generative adversarial network, performing the loss calculation on the pending speech and the to-be-processed speech to obtain an adversarial network loss value of the generative adversarial network; and [0397] performing the loss calculation on the pending speech and the to-be-processed speech to obtain a speech feature loss value, and performing a weighted sum on the adversarial network loss value and the speech feature loss value to obtain the third loss value.

[0398] Through the above method, by acquiring the speech feature vector and the text feature vector, the speech and the text can be received as the inputs, which facilitate the fusion of a TTS task and a VC task to perform multimodal modeling, and improve the performance of the TTS task and the VC task. Furthermore, the speech feature vector and the text feature vector are acquired in the case of a small amount of data, which provides a variety of voice cloning strategies, improves the effect of voice cloning under the small amount of data, reduces the training difficulty and training duration of various models, and supports voice cloning methods in various application scenarios.

[0399] Referring to FIG. **15**, a program product **1500** for implementing the above-mentioned method according to an embodiment of the present disclosure is described. The program product **1500** may take the form of a portable Compact Disc Read-Only Memory (CD-ROM) and include program codes, and may be run on a terminal device, such as a personal computer. However, the program product of the present disclosure is not limited thereto. The readable storage medium herein may be any tangible medium containing or storing a program, and the program may be used by an instruction execution system, an apparatus, or a device, or the program may be used in combination with the instruction execution system, the apparatus, or the device.

[0400] The program product may be any combination of one or more readable mediums.

[0401] The readable medium may be a readable signal medium or a readable storage medium. The readable storage medium may be, for example, but is not limited to, electronic, magnetic, optical, electromagnetic, infrared, or a semiconductor system, apparatus, or device, or any combination thereof. More specific examples (non-exhaustive examples) of readable storage media include: electrical connection with one or more wires, portable disk, hard disk, random access memory (RAM), read-only memory (ROM), erasable programmable read-only memory (EPROM or flash memory), optical fiber, portable compact disc read-only memory (CD-ROM), optical storage device, magnetic storage device, or any suitable combination of the foregoing.

[0402] The computer-readable signal medium may include a data signal in baseband or propagated as part of a carrier wave, which carries readable program codes. Such a propagated data signal may have many forms, including but not limited to electromagnetic signals, optical signals, or any suitable combination of the foregoing.

[0403] The readable signal medium may also be any readable medium other than a readable storage medium, and the readable medium may send, propagate, or transmit a program that is used by an instruction execution system, apparatus, or device, or that is used in combination with an instruction execution system, apparatus, or device.

[0404] The program codes contained on the readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wired, optical fiber, RF, etc., or any suitable combination of the foregoing.

[0405] The program codes for performing the operations of the present disclosure may be written in any combination of one or more programming languages, which include object-oriented programming languages, such as Java, C++, and so on. The programming languages also include conventional procedural programming language, such as “C” or a similar programming language. The program codes may be executed entirely on the user computing device, may be executed partly on the user device, may be executed as an independent software package, may be executed partly on the user computing device and partly on a remote computing device, or may be executed entirely on the remote computing device or server. In the case of a remote computing device, the remote computing device may be connected to the user computing device through any kind of network, including a local area network (LAN) or a wide area network (WAN), or the remote computing device may be connected to an external computing device, for example, by the Internet provided by the Internet service providers.

[0406] Other embodiments of the present disclosure will be apparent to those skilled in the art from consideration of the specification and practice of the present disclosure disclosed herein. The present disclosure is intended to cover any variations, uses, or adaptations of the present disclosure, which are in accordance with the general principles of the present disclosure and include common general knowledge or conventional technical means in the art that are not disclosed in the present disclosure. The specification and embodiments are illustrative, and the real scope and spirit of the present disclosure is defined by the appended claims.

Claims

1. A method for generating a speech, performed by terminal device, comprising: acquiring a speech feature vector of a to-be-processed speech, and inputting the speech feature vector into a speech generation model to obtain a language unit vector; acquiring a text feature vector, and determining a to-be-processed feature vector according to the text feature vector and the language unit vector; and inputting the to-be-processed feature vector into a sequence-to-sequence model to obtain an acoustic feature vector, and inputting the acoustic feature vector into a vocoder to obtain a target speech corresponding to the to-be-processed speech or the text feature vector.

2. The method for generating the speech according to claim 1, wherein inputting the speech feature vector into the speech generation model to obtain the language unit vector comprises: inputting the speech feature vector into the speech generation model to make the speech generation model output a speech encoding vector and a self-restored speech; and performing loss calculation on the to-be-processed speech and the self-restored speech to obtain a first loss value, and determining the speech encoding vector as the language unit vector according to the first loss value.

3. The method for generating the speech according to claim 2, wherein inputting the speech feature vector into the speech generation model to make the speech generation model output the speech encoding vector and the self-restored speech comprises: inputting the speech feature vector into the speech generation model, and performing, by using an encoder module of the speech generation model, nonlinear transformation on the speech feature vector to obtain the speech encoding vector; performing, by using a vector quantization module of the speech generation model, quantization on the speech encoding vector to obtain a speech quantization sequence, and acquiring a speaker vector corresponding to the to-be-processed speech; and performing, by using a decoder module of the speech generation model, the nonlinear transformation on the speech quantization sequence and the speaker vector to obtain the self-restored speech.

4. The method for generating the speech according to claim 3, wherein acquiring the speaker vector corresponding to the to-be-processed speech comprises: acquiring a speaker identity corresponding to the to-be-processed speech, and determining a correspondence between the speaker identity and the speaker vector, wherein the correspondence is determined according to the speech generation model; and querying, according to the correspondence, the speaker vector corresponding to the speaker identity.

5. The method for generating the speech according to claim 3, wherein performing, by using the vector quantization module of the speech generation model, the quantization on the speech encoding vector to obtain the speech quantization sequence comprises: based on a codebook in the vector quantization module of the speech generation model, performing the quantization on the speech encoding vector through a nearest neighbor search algorithm to obtain the speech quantization sequence.

6. The method for generating the speech according to claim 5, wherein performing the quantization on the speech encoding vector through the nearest neighbor search algorithm to obtain the speech quantization sequence comprises: updating the codebook to obtain an updated codebook; and based on the updated codebook, performing the quantization on the speech encoding vector through the nearest neighbor search algorithm to obtain the speech quantization sequence.

7. The method for generating the speech according to claim 6, wherein updating the codebook to obtain the updated codebook comprises: acquiring a codebook identity of each frame of codebook, and comparing codebook identities to obtain a comparison result; and merging codebooks according to the comparison result to obtain the updated codebook.

8. The method for generating the speech according to claim 3, wherein inputting the to-be-processed feature vector into the sequence-to-sequence model to obtain the acoustic feature vector comprises: acquiring a to-be-processed acoustic vector of the to-be-processed feature vector, and inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model to make the sequence-to-sequence model output a processed acoustic vector; and performing the loss calculation on the to-be-processed acoustic vector and the processed acoustic vector to obtain a second loss value, and determining the processed acoustic vector as the acoustic feature vector according to the second loss value.

9. The method for generating the speech according to claim 8, wherein inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model to make the sequence-to-sequence model output the processed acoustic vector comprises: inputting the to-be-processed feature vector and the to-be-processed acoustic vector into the sequence-to-sequence model, and performing, by using an encoder module of the sequence-to-sequence model, nonlinear mapping on the to-be-processed feature vector and the to-be-processed acoustic vector to obtain a spatial encoding vector; summing the spatial encoding vector and the speaker vector to obtain a to-be-aligned vector, and acquiring a speech feature sequence; and aligning, by using an attention mechanism of the sequence-to-sequence model, the to-be-aligned vector and the speech feature sequence to obtain a contextual representation vector, and performing, by using a decoder of the sequence-to-sequence model, the nonlinear mapping on the contextual representation vector to obtain the processed acoustic vector.

10. The method for generating the speech according to claim 1, wherein determining the to-be-processed feature vector according to the text feature vector and the language unit vector comprises: determining the text feature vector or the language unit vector as the to-be-processed feature vector; or summing the text feature vector and the language unit vector to obtain the to-be-processed feature vector.

11. The method for generating the speech according to claim 1, wherein inputting the acoustic feature vector into the vocoder to obtain the target speech corresponding to the to-be-processed speech or the text feature vector comprises: extracting, by a post-processing network, a speech acoustic feature of the acoustic feature vector, and inputting the speech acoustic feature into the vocoder to make the vocoder output a pending speech corresponding to the to-be-processed speech or the text feature vector; and performing loss calculation on the pending speech and the to-be-processed speech to obtain a third loss value, and determining the pending speech as the target speech according to the third loss value.

12. The method for generating the speech according to claim 11, wherein performing the loss calculation on the pending speech and the to-be-processed speech to obtain the third loss value comprises: if the vocoder is a generative adversarial network, performing the loss calculation on the pending speech and the to-be-processed speech to obtain an adversarial network loss value of the generative adversarial network; and performing the loss calculation on the pending speech and the to-be-processed speech to obtain a speech feature loss value, and performing a weighted sum on the adversarial network loss value and the speech feature loss value to obtain the third loss value.

13. (canceled)

14. A non-transitory computer-readable storage medium having a computer program stored thereon, wherein the computer program, when executed by a processor, implements the following operations: acquiring a speech feature vector of a to-be-processed speech, and inputting the speech feature vector into a speech generation model to obtain a language unit vector; acquiring a text feature vector, and determining a to-be-processed feature vector according to the text feature vector and the language unit vector; and inputting the to-be-processed feature vector into a sequence-to-sequence model to obtain an acoustic feature vector, and inputting the acoustic feature vector into a vocoder to obtain a target speech corresponding to the to-be-processed speech or the text feature vector.

15. An electronic device, comprising: a processor; and a memory configured to store executable instructions of the processor; wherein the processor is configured to execute the following operations via executing the executable instructions; acquiring a speech feature vector of a to-be-processed speech, and inputting the speech feature vector into a speech generation model to obtain a language unit vector; acquiring a text feature vector, and determining a to-be-processed feature vector according to the text feature vector and the language unit vector; and inputting the to-be-processed feature vector into a sequence-to-sequence model to obtain an acoustic feature vector, and inputting the acoustic feature vector into a vocoder to obtain a target speech corresponding to the to-be-processed speech or the text feature vector.

16. The method for generating the speech according to claim 1, wherein the speech feature vector of the to-be-processed speech is a Mel-spectrum feature vector.

17. The method for generating the speech according to claim 1, wherein the speech generation model is a Vector Quantization-Variational AutoEncoder (VQ-VAE) model.

18. The method for generating the speech according to claim 1, wherein acquiring the text feature vector comprise: acquiring a natural text, and extracting a phoneme sequence of the natural text as the text feature vector.

19. The method for generating the speech according to claim 9, wherein aligning, by using the attention mechanism of the sequence-to-sequence model, the to-be-aligned vector and the speech feature sequence to obtain the contextual representation vector comprises: performing a dot product calculation on the to-be-aligned vector and the speech feature sequence to obtain the contextual representation vector.

20. The method for generating the speech according to claim 12, wherein performing the weighted sum on the adversarial network loss value and the speech feature loss value to obtain the third loss value comprises: setting corresponding weights for the adversarial network loss value and the speech feature loss value according to empirical values; and performing the weighted sum on the adversarial network loss value and the speech feature loss value to obtain the third loss value.

21. The method for generating the speech according to claim 1, wherein the vocoder is one of a Wavenet model, a Griffin-Lim algorithm, a Generative Adversarial Network (GAN).