

# **A TEXT-TO-SPEECH CONVERSION METHOD, AN APPARATUS, A DEVICE, AND A MACHINE-READABLE MEDIUM**

## **TECHNICAL FIELD**

**[0001]** The present application relates to the speech recognition field and, in particular, to a text-to-speech (TTS) conversion method, an apparatus, a device, and a machine-readable medium.

## **BACKGROUND**

**[0002]** The current available speech synthesis methods are categorized into two types: concatenative synthesis, by which samples of recorded sound of frequently used characters (or words) are concatenated in accordance with the text input; and parametric speech synthesis, by which a constructed speech synthesis model is built to automatically convert text input into an audio file (or spectrum) without the need to concatenate. Deep learning-based speech synthesis technique falls under the category of parametric speech synthesis.

**[0003]** Currently, a challenging problem in the existing speech synthesis techniques is that speech synthesis faces high latency, which is particularly prominent when a long text is input. The high latency is mainly caused by the fact that inference can only be made sequentially based on one time series, not made in parallel, as inference concerning speech synthesis is about calculating time series and the calculation of the next time series relies on the output of the preceding time series. Accordingly, the inference latency that a text input faces in speech synthesis depends on its length. The longer the text input, the higher the latency is.

**[0004]** All existing speech synthesis techniques face this challenge. Inference in both concatenative synthesis and parametric speech synthesis boils down to the calculation of time series. If the simplest approach of concatenative synthesis is deployed, that is, recorded sounds of each character are directly concatenated in accordance with the

text input without regard to context and transitions between characters and between words, then speech synthesis will no longer rely on time series and expedite. However, this will bring many problems: (1) inherent prosody of language will be lost as it generates unnatural word-by-word speech; (2) polyphones cannot be processed as each polyphonic character only retains a default pronunciation without regard to context. Therefore, as the quality of synthetic speech it generates can't be guaranteed, especially in regard to naturalness, this method is not ideal in spite of low-latency inference.

[0005] There are many deep learning models suitable for parallel inference. However, the mainstream speech synthesis model is based on recurrent neural networks, to which parallel inference does not apply.

## **SUMMARY**

[0006] The purposes of the present application are to provide a text-to-speech (TTS) conversion method, an apparatus, a device, and a machine-readable medium and to improve the accuracy and efficiency of TTS conversion.

[0007] First, the present application provides a TTS conversion method comprising:

[0008] Obtaining text which is to be segmented into character substrings, and in two adjacent character substrings, the word at the end of the preceding character substring is identical to the word at the beginning of the next character substring;

[0009] Generating audio units corresponding to the character substrings, and in the audio units corresponding to two adjacent character substrings, similar audio frames are existing between the audio frames at the end of the preceding audio unit and the audio frames at the beginning of the next audio unit;

[0010] Concatenating the beginning of an audio unit to the end of the preceding audio unit from the location where both audio units have similar audio frames, thus generating speech corresponding to the two adjacent character substrings.

[0011] In a possible representation of the first aspect above, segmenting text into at least one character substring comprises: based on a predefined length of character, the length of character substrings after segmentation is equal to the predefined length of

character sequences.

**[0012]** In a possible representation of the first aspect above, generating audio units corresponding to the character substrings comprises: inputting character substrings into a neural network identification model which identifies audio units corresponding to the character substrings.

**[0013]** In a possible representation of the first aspect above, the text is in Chinese and is to be segmented using Chinese word segmentation methods.

**[0014]** In a possible representation of the first aspect above, based on a predetermined parallel threshold, character substrings, whose quantity is equivalent to the parallel threshold value, are converted into corresponding audio units through parallel conversion.

**[0015]** Second, the present application provides a TTS conversion apparatus comprising:

**[0016]** A text segmentation module, which is for obtaining text that is to be segmented into character substrings, and in two adjacent character substrings, the word at the end of the preceding character substring is identical to the word at the beginning of the next character substring;

**[0017]** An audio conversion module, which is for generating audio units corresponding to the character substrings, and in the audio units corresponding to two adjacent character substrings, similar audio frames are existing between the audio frames at the end of the preceding audio unit and the audio frames at the beginning of the next audio unit;

**[0018]** An audio synthesis module, which is for concatenating the beginning of an audio unit to the end of the preceding audio unit from the location where both audio units have similar audio frames, thus generating speech corresponding to the two adjacent character substrings.

**[0019]** In a possible representation of the second aspect above, segmenting text into at least one character substring by a text segmentation module comprises: based on a predefined length of character, the length of character substrings after segmentation is equal to the predefined length of character sequences.

**[0020]** In a possible representation of the second aspect above, generating audio units corresponding to the character substrings by an audio conversion module comprises: inputting, by the audio conversion module, character substrings into a neural network identification model which identifies audio units corresponding to the character substrings.

**[0021]** In a possible representation of the second aspect above, based on a predetermined parallel threshold, character substrings, whose quantity is equivalent to the parallel threshold value, are generated by an audio synthesis module and an audio conversion module into corresponding audio units through parallel conversion.

**[0022]** Third, the present application provides a TTS conversion device comprising:

**[0023]** A memory that stores instructions, and a processor which is for enabling the device to read and execute the instructions stored in the memory, thereby implementing any of the method for representations as described in the first aspect above.

**[0024]** Fourth, the present application provides a machine-readable medium, wherein stored instructions are executed by a machine that implements any of the method for representations as described in the first aspect above.

**[0025]** The embodiments of the present application provide a TTS conversion method. A text is segmented into multiple character substrings where adjacent ones overlap, which are then converted into audio units through parallel conversion. During the process of conversion, character substrings are concatenated with the help of the overlaps of adjacent character substrings, thus improving the efficiency and accuracy of TTS conversion.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

**[0026]** FIG. 1 is a flowchart describing the text-to-speech (TTS) conversion method presented by the embodiments of the present application.

**[0027]** FIG. 2 illustrates a scenario describing the TTS conversion process presented by the embodiments of the present application.

**[0028]** FIG. 3 is a structure diagram illustrating the identification model for the TTS

conversion process presented by the embodiments of the present application.

[0029] FIG. 4 is a structure diagram illustrating the identification model for the TTS conversion process presented by the embodiments of the present application.

[0030] FIG. 5 illustrates a scenario describing the TTS conversion process presented by the embodiments of the present application.

[0031] FIG. 6 is a structure diagram illustrating the modules for the TTS conversion process presented by the embodiments of the present application.

[0032] FIG. 7 is a schematic diagram illustrating the computing system presented by some embodiments of the present application.

[0033] FIG. 8 is a block diagram illustrating a system-on-a-chip (SoC) presented by the embodiments of the present application.

## **DESCRIPTION OF THE EMBODIMENTS**

[0034] The following describes the technical solutions involved in the embodiments of the present application in conjunction with the accompanying drawings. As used herein, “/” indicates or, unless specifically stated otherwise. For example, A/B may indicate A or B. The term “and/or” used herein merely describes three relationships between associated items. For example, A and/or B may indicate three scenarios: A exists alone, A and B exist simultaneously, and B exists alone. Further, the term “multiple” used herein is intended to refer to two or more than two items.

[0035] For a more complete understanding of the purposes, technical solutions and advantages of and involved under the present application, the following provides a further description of the implementation methods for the present application in conjunction with the accompanying drawings.

[0036] As per some embodiments of the present invention, the letter behind a reference number as used in FIG. 1 and other drawings, for example, “100a,” indicates a reference to the element with the specific reference number, whereas a reference number without a letter that follows behind, for example, “100,” indicates a general reference to the implementation method for the element with the reference number.

**[0037]** The embodiments of the present invention provide a TTS conversion method. A text is segmented into multiple character substrings where adjacent ones overlap, which are then converted into audio units through parallel conversion. During the process of conversion, character substrings are concatenated with the help of the overlaps of adjacent character substrings, thus improving the efficiency and accuracy of TTS conversion. This method may be performed using a computer device which may be any device capable of calculation, such as personal computer, laptop, personal workstation or server.

**[0038]** FIG. 1 is a flowchart describing the TTS conversion method presented by an example embodiment. This parallel speech synthesis method may comprise the following steps as shown in FIG. 1.

**[0039]** S101: Obtain a text input and segment it into at least one character substring of similar length.

**[0040]** In the embodiments of the present invention, take Chinese text input as an example. Sequences of characters in the Chinese writing system are usually separated using punctuation marks, while the English writing system separates words with spaces. Therefore, the fundamental task in natural language processing dealing with the Chinese language is to segment Chinese characters sequences into words. Chinese word segmentation refers to the task of splitting Chinese text (i.e. Chinese characters) into one or multiple words, that is, using spaces to separate words like English.

**[0041]** We hereby introduce the MMSEG algorithm. A problem in computation analysis of Chinese text is that there are no word boundaries in conventionally printed text. Since a word is such a fundamental linguistic unit, it is necessary to identify words in Chinese text, that is, to perform Chinese word segmentation, so that higher-level analyses can be performed. Specifically, the MMSEG algorithm identifies multiple combinations of characters as words in a complete sentence from left to right, helping to solve the problem of only considering the word itself and ignoring the context-related words during the process of selecting words.

**[0042]** In the embodiments of the present invention, text input can be segmented into words based on a predetermined substring length in combination with the MMSEG

algorithm above. Character strings are segmented at word boundaries based on word segmentation outcomes to ensure substrings obtained after segmentation are of similar length which conforms to the predefined value. The length here may be inferred based on the latency requirement for speech synthesis. If the speech synthesis speed is  $n$  characters per second, the latency requirement is  $m$  seconds, and the length requirement for a substring is  $n*m$  characters. That is, the length of a substring that can be processed within  $m$  seconds is  $n*m$ . The latency requirement here refers to the time during which the computing devices, which implement the method as described in the embodiments of the present invention, need to respond. If a text input consisting of  $L$  characters is given, then approximately  $L/(n*m)$  substrings will be generated.

**[0043]** Furthermore, in the embodiments of the present invention, parallel segmentation of text input into substrings and conversion of text to speech can also be performed. In a circumstance where the CPU of the computing device that implements the TTS conversion method of the present invention has  $N$ , an integer greater than 1, cores, then the number of substrings that can be produced through parallel text segmentation is also  $N$ .

**[0044]** The segmentation of a long text into several substrings also enables the alignment of the characters in adjacent substrings. That is, the ending word of the preceding substring can be aligned with the first word of the next substring, thus ensuring consistency with the original text after adjoining two adjacent substrings. Despite that the substrings are aligned, it is hard to ensure that audio units generated based on the substrings can be adjoining naturally, and characters are even missed or skipped since text boundaries can be uncommon during the segmentation of text input and the speech synthesis model is unable to produce audio units with natural beginnings and endings for unfamiliar characters/words that act as boundaries. In the embodiments of the present invention, an additional substring between two adjacent substrings is added, as a result of which the words contained in the additional substring overlap with the end of the preceding substring and the start of the next substring.

**[0045]** For example, as shown in FIG. 2, “A lone boat, a stranded boy” in “A lone boat, a stranded boy, a Bengal tiger, and the toughest test to survive in the South Pacific.” is segmented into two substrings, i.e. “A lone boat” and “a stranded boy” when the predefined substring length is 6. Meanwhile, an additional substring “boat, a stranded” between the two substrings is provided.

**[0046]** This is to produce overlaps between substrings. In the subsequent step of generating audio units, the audio units corresponding to the preceding and the next substrings can be concatenated from where they overlap. In this step, the substrings after segmentation are not necessarily of identical length. While segmentation is performed in accordance with the predefined substring length, the outcomes of word segmentation are also taken into consideration, making the boundaries of text segmentation fall on the boundaries of segmented words to ensure lexical integrity.

**[0047]** S102: Convert text substrings into corresponding speech to generate audio unit files corresponding to the substrings.

**[0048]** Different methods may be employed for the conversion of substrings into corresponding speech. The present invention, instead of being limited to a single TTS conversion algorithm, applies to different existing TTS conversion techniques.

**[0049]** In the embodiments of the present invention, take the Google Tacotron 2 model shown in FIG. 3 as an example. This model uses sequences of character vectors as input to output spectrogram frames and then generates corresponding audio via the Griffin-Lim algorithm.

**[0050]** The G&L (Griffin-Lim) algorithm mentioned here is a method for reconstructing speech waveforms through a given amplitude spectrogram and an unknown phase spectrogram which is to be calculated via iteration. This algorithm converts acoustic parameters generated by the speech synthesis system into speech waveforms and estimates phase information based on the relationships between audio frames, thereby reconstructing speech waveforms.

**[0051]** Each substring after segmentation is converted into a corresponding vector which will then be input to a convolutional neural network (CNN) to generate an audio unit, a process to which the encoder-decoder architecture of Tacotron2 training

model is applied. A recurrent neural network (RNN) can be embedded into the encoder to encode character strings, and long short-term memory (LSTM) can be embedded into the decoder to predict training data. In the end, the CNN embedder in the encoder of the Tacotron2 training model will convert the vector data generated from text selection into vectors. The LSTM embedded in the training model's decoder will combine the content generated by the RNN embedded in the encoder with the substrings' vector to input and produce audio units.

**[0052]** S103: During the process of converting text substrings into corresponding audio units, concatenate the audio units corresponding to the substrings to generate complete speech corresponding to the text.

**[0053]** In the embodiments of the present invention, the overlaps of two adjacent audio units converted from two overlapping text substrings will be compared. When the similarity between the ending audio frames of the audio unit for the preceding character substring and the beginning audio frames of the audio unit for the next character substring is higher than a threshold value, it will be determined that the two substrings overlap with each other to some extent and they will be concatenated from where they overlap. Furthermore, in the embodiments of the present invention, parallel concatenation of multiple substrings can also be performed. In a circumstance where the CPU of the computing device that implements the TTS conversion method of the present invention has N, an integer greater than 1, cores, then the number of substrings that can be produced through parallel text segmentation is also N.

**[0054]** For example, as shown in FIG. 4, after the segmentation described in the S101 step, when audio conversion for the substring "a lone boat" proceeds from "a" to "boat" and that for the substring "boat, a stranded" proceeds to "boat," if the similarity between the beginning audio frames of "boat" in "a lone boat" and the beginning audio frames of "boat" in "boat, a stranded" is higher than a threshold value, then the audio units generated for "a lone boat" and "boat, a stranded" will be concatenated into an audio file from the audio frames which have a similarity higher than the threshold value. That is, an audio file for "a lone boat, a stranded" will be generated. Likewise, when an audio unit for "a stranded boy" is being generated, if

the similarity between the beginning audio frames of “a” in “boat, a stranded” and the beginning audio frames of “a” in “a stranded boy” is higher than a threshold value, then the audio units generated for “boat, a stranded” and “a stranded boy” will be concatenated into an audio file from the audio frames which have a similarity higher than the threshold value. That is, an audio file for “boat, a stranded boy” will be generated. In a circumstance where 3 substrings can be processed in parallel, audio units for the three substrings, i.e. “a lone boat,” “a stranded boy,” and “boat, a stranded”, will be generated simultaneously. Therefore, the synthesis of the audio units for “a lone boat” and “boat, a stranded” and the synthesis of the audio units for “boat, a stranded” and “a stranded boy” will be performed simultaneously. In the end, an audio file for “a lone boat, a stranded boy” will be obtained.

**[0055]** The following introduces how to determine whether the audio frames contained in different audio units are similar or not. In the following sections,  $w[i]$  is intended to represent the audio unit corresponding to substring  $i$  and  $w[i][j]$  is intended to represent audio frame  $j$  in the audio unit, and both “ $i$ ” and “ $j$ ” used here are integers.

**[0056]** As shown in FIG. 5, both the preceding and the next audio units include “boat”, so find out the starting location of audio frames of “boat” contained in the preceding audio unit, i.e. the first audio unit. Obtain the starting location  $w[i][j]$  of the audio corresponding to “boat.” That is, find the beginning audio frame  $w[i][j]$  of the audio corresponding to “boat.”

**[0057]** Then, find the overlaps between the two audio units. Traverse the audio frames contained in the preceding audio unit from  $w[i][j]$  and compare them with the audio frames beginning from  $w[i+1][1]$  (assuming it starts from 1) included in the next audio unit. If the similarity between  $w[i][j] \dots w[i][k]$  and  $w[i+1][1] \dots w[i+1][h]$  is higher than the threshold value, it will be confirmed that the two audio portions,  $w[i][j] \dots w[i][k]$  and  $w[i+1][1] \dots w[i+1][h]$ , coincide. In a circumstance where  $k-j$  equates to  $h-1$ , it indicates that the overlaps of  $w[i]$  and  $w[j]$  are completely identical. Many methods are available for determining the similarity between two audio frames, such as the cosine similarity algorithm and the Euclidean distance algorithm.

**[0058]** Last, concatenate the two audio units, i.e.  $w[i]$  and  $w[i+1]$ , from the location of  $w[i][k]$  and  $w[i+1][h+1]$  to generate a complete audio file for the preceding and the next audio units.

**[0059]** FIG. 6 is a structural diagram illustrating the TTS conversion apparatus presented by an example embodiment. As shown in FIG. 6, the speech synthesis apparatus may comprise a text segmentation module 601, an audio conversion module 602, and an audio synthesis module 603.

**[0060]** The text segmentation module 601 is used for obtaining a text input and segmenting it into at least one character substring of similar length.

**[0061]** The audio conversion module 602 is used for converting character substrings into corresponding speech to generate audio units corresponding to the substrings.

**[0062]** The audio synthesis module 603 is used for concatenating audio units corresponding to character substrings during the process of converting the substrings after text segmentation into corresponding audio units to generate complete speech corresponding to the text.

**[0063]** The following illustrates the example computing system 700 presented in some embodiments of the present application in conjunction with FIG. 7. The example computing system 700 may be represented in the TTS conversion method or apparatus as described above in conjunction with FIG. 1 through FIG. 6.

**[0064]** In one embodiment, the system 700 may include one or more processors 704, a system control logic 708 connected to at least one of the processors 704, a system memory 712 connected to the system control logic 708, a memory 716 connected to the system control logic 708 (for example, a non-volatile memory [NVM]), and a network interface 720 connected to the system control logic 708.

**[0065]** The processors 704 may include one or more single-core or multi-core processors, and may include any combination of general processors and specialized processors (for example, graphics processors, application processors, and baseband processors). In some circumstances, the processors 704 may be configured to implement any of the TTS conversion operations as described above in conjunction with FIG. 1 through FIG. 5.

**[0066]** The system control logic unit 708 used for some embodiment may include any appropriate interface controller to provide appropriate interface for at least one of the processors 704 and/or any appropriate device or component communicating with the system control logic 708.

**[0067]** The system control logic unit 708 used for some embodiment may include one or multiple memory controllers to provide interface connected to the system memory 712. The system memory 712 may be used for loading and storing data and/or instructions. For example, for the system 700, the system memory 712 used for some embodiment may include any appropriate volatile memory, for example, appropriate random-access memory (RAM) or dynamic random access memory (DRAM).

**[0068]** The memory 716 may include one or more tangible, non-transitory computer readable media used for storing data and/or instructions. For example, the memory 716 may include any appropriate non-volatile memory and/or non-volatile storage device, for example, flash memory, hard disk drive (HDD), solid-state drive (SSD), compact disk (CD), and/or digital versatile disk (DVD) drive.

**[0069]** The memory 716 may include some of the storage resources of the apparatus installed on the system 700, or may be accessed by devices yet not necessarily a part of the devices. For example, the memory 716 may be accessed online via the network interface 720.

**[0070]** In particular, the system memory 712 and the memory 716 may each include: temporary and permanent copies of the instructions 724. The instructions 724 may include: instruction executed by at least one of the processors 704 to make the system 700 implement any of the method as described above. In a variety of embodiments, the instructions 724 or hardware, firmware, and/or software may be placed within the system control logic 708, the network interface 720 and/or the processors 704 as supplements/substitutes.

**[0071]** The network interface 720 may include a transceiver for providing radio interface to the system 700 to allow it to communicate with any appropriate device (for example, front-end modules and antennas) via one or multiple networks. In a variety of embodiments, the network interface 720 may be integrated with other

components of the system 700. For example, the network interface may include the processors of the processors 704, the memory of the system memory 712, the memory of the memory 716, and/or firmware (not indicated) with at least one instruction which is to be executed by at least one of the processors 704 to make the system 700 implement any of the TTS conversion methods as described in FIG. 1 through FIG. 5.

**[0072]** The network interface 720 may further include any appropriate hardware and/or firmware to provide multiple-input multiple-output radio interface. For example, the network interface 720 used for some embodiment may be a network adapter, a wireless network adapter, a telephone modem, and/or a wireless modem.

**[0073]** In one embodiment, at least one of the processors 704 may be enclosed with the logic of one or more controllers used for the system control logic 708. In one embodiment, at least one of the processors 704 may be enclosed with the logic of one or more controllers used for the system control logic 704 to form a system in a package (SiP). In one embodiment, at least one of the processors 704 may be integrated with the logic of one or more controllers used for the system control logic 708. In one embodiment, at least one of the processors 704 may be integrated with the logic of one or more controllers used for the system control logic 708 to form a system on a chip (SoC).

**[0074]** The system 700 may further include: an input/output (I/O) device 732. The I/O device 732 may include: a user interface designed to enable users to interact with the system 700; a peripheral component interface designed to enable peripheral components to interact with the system 700; and/or sensors designed to determine environment conditions and/or location information related to the system 700.

**[0075]** In a variety of embodiments, the user interface may include, without limitation, monitors (for example, LCD monitors and touch screen monitors), speakers, microphones, one or multiple cameras (for example, still video cameras and/or video cameras), flashlight/flash (for example, LED flash), and keyboards.

**[0076]** In a variety of embodiments, the peripheral component interface may include, without limitation, non-volatile memory ports, audio jacks, and power interfaces.

**[0077]** In a variety of embodiments, the sensors may include, without limitation,

gyroscope sensors, accelerometers, proximity sensors, ambient light sensors, and positioning units. Positioning units may be a part of the network interface 720 or interact with the network interface 720 to communicate with the components of positioning networks (for example, global positioning network [GPS] satellites).

**[0078]** FIG. 8 is a block diagram illustrating a system-on-a-chip (SoC) 800 in accordance with the embodiments of the present application. In FIG. 8, the dotted frame is an optional feature of the SoC. In FIG. 8, the SoC 800 may include: an interconnect unit 850 which is coupled with the application processor 810; a system proxy unit 870; a bus controller unit 780; an integrated memory controller unit 840; a set of or one or more coprocessors 820 which may include integrated graphics logic, graphics processors, audio processors and video processors; a static random-access memory (SRAM) unit 830; and a direct memory access (DMA) unit 860. In one embodiment, the coprocessors 820 may include specialized processors, such as network or communications processors, compression engines, GPGPU, high-throughput MIC processors, and built-in processors.

**[0079]** The embodiments disclosed in the present application may be represented in hardware, software, firmware, or combinations of these representation methods. The embodiments of the present application may be represented as computer programs or program code executed on a programmable system which may include at least one processor, a storage system (including volatile memory and non-volatile memory and/or memory cells), at least one input device, and at least one output device.

**[0080]** The program code may be applied to input instructions in order to execute the functions described in the present application and generate output information. The output information may be applied to one or more output devices by known methods. For the purposes of the present application, processing systems include any system with processors such as digital signal processors (DSP), microcontrollers, application-specific integrated circuits (ASIC), and microprocessors.

**[0081]** The program code may be represented using high-level programming languages or object-oriented programming languages to communicate with the processing systems. Assembly languages or machine languages may also be used for

the representation of the program code when necessary. In fact, the mechanisms described in the present application are not limited to any specific programming language. In any circumstance, the programming language may be a compiled language or an interpreted language.

**[0082]** In some circumstances, the embodiments disclosed may be represented through hardware, firmware, software, or any of their combinations. The embodiments disclosed may also be represented through instructions or programs carried by or stored in one or more transient or non-transient machine-readable (for example, computer-readable) media, which may be read or executed by one or more processors. When the instructions or programs are run by a machine, the machine may implement any of the methods described above. For example, the instructions may be distributed via networks or other computer-readable media. Therefore, machine-readable media may include, without limitation, any medium capable of storing or transmitting information in a format readable by a machine (for example, computers), for example, floppy disks, optical discs, compact disc read-only memory (CD-ROMs), magneto-optical discs, read-only memory (ROM), random access memory (RAM), erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), magnetic and optical cards, and flash memory or tangible machine-readable memory that transmits network information via electrical, optical, sound, or other form of signals (for example, carrier waves, infrared signals, and digital signals). Further, machine-readable media include any form of machine-readable medium suitable for storing or transmitting electronic instructions or information readable by machines (for example, computers).

**[0083]** Therefore, the embodiments of the present application also include non-transient, tangible machine-readable media which include instructions or design data, such as hardware description languages (HDL) that define the structures, electronic circuits, apparatus, processors and/or system features described in the present disclosure. These embodiments are also referred to as program products.

**[0084]** Although a detailed description is made on the embodiments of the present application with reference to the accompanying drawings, it is to be noted that the

uses of the technical solutions in the present application are not limited to the various applications as described in the embodiments and that many structures and variations may be implemented with reference to the technical solutions of the present application in order to achieve the benefits and advantages of the present disclosure. Various changes that are known to those skilled in the art and are not beyond the purposes of the present application are to be understood as being included within the scope of the present patent application.

## Claims

### WHAT IS CLAIMED IS:

1. A text-to-speech (TTS) conversion method, comprising:  
obtaining text which is to be segmented into character substrings, each character substring comprises multiple words, and in two adjacent character substrings, the word at the end of the preceding character substring is identical to the word at the beginning of the next character substring, said text is in Chinese and is to be segmented using Chinese word segmentation methods;  
generating audio units corresponding to said character substrings, and in the audio units corresponding to two adjacent character substrings, similar audio frames are existing between the audio frames at the end of the preceding audio unit and the audio frames at the beginning of the next audio unit;  
concatenating the beginning of an audio unit to the end of the preceding audio unit from the location where both audio units have similar audio frames, thus generating speech corresponding to the two adjacent character substrings.
2. The method described in Claim 1, wherein said text is segmented into at least one character substring comprising: based on a predefined length of character, the length of character substrings after segmentation is equal to the predefined length of character sequences.
3. The method described in Claim 1, wherein generating audio units corresponding to said character substrings comprises: inputting the character substrings into a neural network identification model which identifies audio units corresponding to the character substrings.
4. The method described in Claim 1, wherein based on a predetermined parallel threshold, character substrings, whose quantity is equivalent to the said parallel threshold value, are converted into corresponding audio units through parallel conversion.
5. A TTS conversion apparatus comprising:

a text segmentation module, which is for obtaining text to be segmented into character substrings, each character substring comprises multiple words, and in two adjacent character substrings, the word at the end of the preceding character substring is identical to the word at the beginning of the next character substring, said text is in Chinese and is to be segmented using Chinese word segmentation methods;

an audio conversion module, which is for generating audio units corresponding to said character substrings, and in the audio units corresponding to two adjacent character substrings, similar audio frames are existing between the audio frames at the end of the preceding audio unit and the audio frames at the beginning of the next audio unit;

an audio synthesis module, which is for concatenating the beginning of an audio unit to the end of the preceding audio unit from the location where both audio units have similar audio frames, thus generating speech corresponding to the two adjacent character substrings.

6. The apparatus described in Claim 5, wherein a text segmentation module segments said text into at least one character substring comprising: based on a predefined length of character, the length of character substrings after segmentation is equal to the predefined length of character sequences.

7. The apparatus described in Claim 5, wherein said audio conversion module generate audio units corresponding to said character substrings, the apparatus comprising: inputting, by the audio conversion module, the character substrings into a neural network identification model which identifies audio units corresponding to the character substrings.

8. The apparatus described in Claim 5, wherein based on a predetermined parallel threshold, character substrings, whose quantity is equivalent to the said parallel threshold value, are generated by said audio synthesis module and said audio conversion module into corresponding audio units through parallel processing.

9. A device, comprising:

a memory that stores instructions, and a processor which is for enabling said device to read and execute the instructions stored in said memory, thereby implementing any of the method as described in Claims 1 through 4.

10. A machine-readable medium, wherein stored instructions are executed by a machine that implements any of the method as described in Claims 1 through 4.