

SPEECH WAKEUP METHOD, APPARATUS, AND ELECTRONIC DEVICE

Technical Field

This specification relates to the field of computer software technologies, and in particular,
5 to a speech wakeup method, apparatus, and electronic device.

Technical Background

With the rapid development of mobile devices, speech-related technologies are also becoming increasingly common. For example, speech recognition is used in increasingly
10 popular conversational assistants like Apple's Siri, Microsoft's Cortana, and Amazon's Alexa to enhance user experience and natural human-computer interaction level.

An important speech interaction technology is Keyword Spotting (KWS), which may also be generally referred to as speech wakeup. Based on the prior art, there is a need for a speech wakeup solution that does not rely on keyword-specific speech data.

15

Summary of the Invention

A speech wakeup method, apparatus, and electronic device are provided in embodiments of this specification, for solving the following technical problem: there is a need for a speech wakeup solution that does not rely on keyword-specific speech data.

20 In order to solve the foregoing technical problem, the embodiments of this specification are implemented as follows.

A speech wakeup method provided in an embodiment of this specification comprises:
inputting speech data to a speech wakeup model trained with general speech data, and
outputting, by the speech wakeup model, a result for determining whether to execute speech

wakeup, wherein the speech wakeup model includes a Deep Neural Network (DNN) and a Connectionist Temporal Classifier (CTC).

A speech wakeup apparatus provided in an embodiment of this specification comprises an input module and a speech wakeup model;

5 wherein speech data is input by the input module to the speech wakeup model trained with general speech data, and the speech wakeup model outputs a result for determining whether to execute speech wakeup, wherein the speech wakeup model includes a DNN and a CTC.

An electronic device provided in an embodiment of this specification comprises:

10 at least one processor; and

a memory communicatively connected to the at least one processor; wherein

the memory stores an instruction executable by the at least one processor, and the instruction is executed by the at least one processor to enable the at least one processor to:

input speech data to a speech wakeup model trained with general speech data, and output,

15 by the speech wakeup model, a result for determining whether to execute speech wakeup, wherein the speech wakeup model includes a DNN and a CTC.

At least one of the foregoing technical solutions adopted in the embodiments of this specification can achieve the following beneficial effects: instead of relying on keyword-specific speech data, the speech wakeup model may be trained with accessible general
20 speech data, and then the trained speech wakeup model can be used for speech wakeup, which is conducive to improving the accuracy of speech wakeup.

Brief Description of the Drawings

In order to describe the technical solutions in the embodiments of this specification or in
25 the prior art more clearly, the accompanying drawings to be used in the description about the

embodiments or the prior art will be briefly introduced in the following. It is apparent that the accompanying drawings described in the following are merely some embodiments disclosed in this specification. Those of ordinary skill in the art may further obtain other accompanying drawings according to these accompanying drawings without creative efforts.

5 FIG. 1 is a schematic diagram of an overall architecture involved in a solution of this specification in an actual application scenario;

FIG. 2 is a schematic flowchart of a speech wakeup method according to an embodiment of this specification;

10 FIG. 3 is a schematic diagram of a framework of the speech wakeup model in FIG. 2 according to an embodiment of this specification;

FIG. 4 is a schematic diagram of feature extraction of the feature extracting module in FIG. 3 in an actual application scenario according to an embodiment of this specification;

FIG. 5 is a schematic structural diagram of a Deep Neural Network (DNN) in FIG. 3 in an actual application scenario according to an embodiment of this specification;

15 FIG. 6 is a schematic structural diagram of a Connectionist Temporal Classifier (CTC) in FIG. 3 in an actual application scenario according to an embodiment of this specification; and

FIG. 7 is a schematic structural diagram of a speech wakeup apparatus corresponding to FIG. 2 according to an embodiment of this specification.

20

Detailed Description

A speech wakeup method, apparatus, and electronic device are provided in the embodiments of this specification.

25 In order to enable those skilled in the art to better understand the technical solutions in this specification, the technical solutions in the embodiments of this specification will be

clearly and completely described in the following with reference to the accompanying drawings in the embodiments of this specification. It is apparent that the embodiments described are merely some, rather than all, of the embodiments of this application. All other embodiments obtained by those of ordinary skill in the art based on the embodiments of this specification without creative efforts should all fall within the protection scope of this application.

For ease of understanding, the idea of the solutions of this specification is explained as follows. In this specification, a speech wakeup model including a Deep Neural Network (DNN) and a Connectionist Temporal Classifier (CTC) is trained with general speech data.

The trained speech wakeup model can be used for speech wakeup and support user-defined keywords triggered by speech wakeup. Further, the speech wakeup model can be used in low-power devices such as mobile phones and home appliances, because the DNN included in the speech wakeup model can be relatively not too complicated, and for example, can only have three or four layers with two or three hundred nodes in each layer. The speech wakeup model can be referred to as CTC-KWS, and the KWS here is the keyword spotting mentioned in the Background.

The DNN is a multi-layer perceptron, which has a hidden layer between an input layer and an output layer and can simulate complex nonlinear relationships. The CTC is a classifier configured to perform a label tagging task, and does not require forced alignment between input and output.

FIG. 1 is a schematic diagram of an overall architecture involved in a solution of this specification in an actual application scenario. Two parts are mainly involved in the overall architecture: speech data and a speech wakeup model. The speech wakeup model includes a DNN and a CTC. Speech wakeup can be implemented by inputting the speech data to the speech wakeup model for processing.

The solution of this specification is described in detail in the following based on the foregoing idea and overall architecture.

FIG. 2 is a schematic flowchart of a speech wakeup method according to an embodiment of this specification. From the perspective of programs, the executive body of the flow may be a program on a server or a terminal, for example, a model training program, a speech recognition program, a speech wakeup application, and so on. From the perspective of devices, the possible executive body of the flow is, but not limited to, at least one of the following devices that can serve as servers or terminals: a mobile phone, a tablet computer, a smart wearable device, an automobile machine, a personal computer, a medium-sized computer, a computer cluster, and so on.

The flow in FIG. 2 may include the following steps.

In S202, speech data is input to a speech wakeup model trained with general speech data.

In the embodiment of this specification, speech can be monitored by the executive body or another body to obtain the speech data. When the speech is monitored, a user can speak out a predetermined keyword to trigger the speech wakeup model to execute speech wakeup.

In S204, the speech wakeup model outputs a result for determining whether to execute speech wakeup, wherein the speech wakeup model includes a DNN and a CTC.

In the embodiment of this specification, compared with the keyword-specific speech data mentioned in the Background, the general speech data described in step S202 is less restricted and, thus, easily accessible. For example, it may be a Large Vocabulary Continuous Speech Recognition (LVCSR) corpus or the like.

In the embodiment of this specification, the DNN included in the speech recognition model may predict a posterior probability distribution of a pronunciation phoneme sequence corresponding to input speech features. The DNN can be followed by the CTC to give a confidence score corresponding to the predicted pronunciation phoneme sequence. A result

for determining whether to execute speech wakeup can be output based on the confidence score.

With the method in FIG. 2, instead of relying on keyword-specific speech data, the speech wakeup model may be trained with the accessible general speech data, and further
5 the trained speech wakeup model can be used for speech wakeup, which is conducive to improving the accuracy of speech wakeup.

The speech wakeup model also breaks through the restrictions of the keyword-specific speech data and supports user-defined triggered keywords. Therefore, it is more convenient and flexible in actual applications and conducive to improving user experience.

10 Based on the method in FIG. 2, some specific implementation solutions and expanded solutions of the method are further provided in the embodiments of the specification, which are described in the following.

For ease of understanding, a schematic diagram of a framework of the speech wakeup model in FIG. 2 is provided in an embodiment of this specification, as shown in FIG. 3.

15 The framework in FIG. 3 includes a feature extracting module, a DNN, and a CTC in sequence. In step S204, the outputting, by the speech wakeup model, a result for determining whether to execute speech wakeup may specifically include:

extracting acoustic features from the input speech data;

inputting the acoustic features to the DNN for processing to obtain a class probability of
20 the acoustic features respectively corresponding to each pronunciation phoneme;

inputting the class probability to the CTC for processing to obtain a confidence score of a speech wakeup term corresponding to a pronunciation phoneme sequence; and

determining whether to execute wakeup according to the confidence score, and outputting a determination result.

Various parts of the framework in FIG. 3 are described in detail in further combination with FIG. 4, FIG. 5, and FIG. 6 according to the foregoing flow of speech wakeup.

FIG. 4 is a schematic diagram of feature extraction of the feature extracting module in FIG. 3 in an actual application scenario according to an embodiment of this specification.

5 In FIG. 4, when a section of speech "Zhi Ma Kai Men" is monitored at present, a target label sequence corresponding thereto is a pronunciation phoneme sequence, which can be expressed as: "zhi1ma2kai1men2", wherein the numbers represent tones. In addition to phonemes such as initials and finals, tone phonemes are also taken into account as a modeling unit. In an actual application, context-independent or context-dependent phonemes
10 can all be taken into account, among which the latter is more numerous. However, in consideration of reducing subsequent computational burden of the DNN, it is preferable to consider only the context-independent phonemes, specifically 72 context-independent Chinese phoneme units, including a blank unit.

Acoustic features can be extracted by the feature extracting module from the input speech
15 data, which may specifically include extracting acoustic feature frames of the input speech data from a window according to a specified time interval, wherein each of the acoustic feature frames is multi-dimension log filter bank energies; stacking a plurality of adjacent acoustic feature frames respectively; taking the stacked acoustic feature frames respectively as acoustic features extracted from the general speech data; and further, the stacked acoustic
20 feature frames can be used as inputs of the DNN respectively.

The log filter bank energies refer to energy signals extracted by a log filter bank, which can be expressed as a vector in the solution of this specification to facilitate model processing. The multi-dimension in the foregoing represents multiple dimensions of the vector.

For example, a specified length of a time window may be 25 milliseconds, each time
25 window may move for 10 milliseconds, and the multi-dimension may be, for example, 40

dimensions. On a time axis of the speech data, milliseconds from 0 to 25 may be used as a window, and 40-dimension log filter bank energies are correspondingly extracted from the speech data to serve as a first acoustic feature frame; milliseconds from 10 to 35 can be used as a window, and 40-dimension log filter bank energies are correspondingly extracted from the speech data to serve as a second acoustic feature frame; and multiple acoustic feature frames can be extracted in the same way.

Further, the objective of stacking a plurality of adjacent acoustic feature frames is to allow more information from a context of a current frame, which is conducive to improving the accuracy of subsequent prediction results. Following the above example, the current frame, the adjacent consecutive ten frames before the current frame, and the adjacent consecutive five frames after the current frame can be for example stacked to obtain a 640-dimension stacking feature for being inputted to the subsequent DNN. In addition, cepstral mean and variance normalization can be carried out on the dimensions of the stacking feature, and then backward input can be carried out.

It should be noted that the feature extracting manner and parameters adopted in the foregoing example are only exemplary, and they can be adjusted as needed in actual applications.

FIG. 5 is a schematic structural diagram of the DNN in FIG. 3 in an actual application scenario according to an embodiment of this specification.

In FIG. 5, various neurons in the DNN are fully connected. The acoustic features extracted by the feature extracting module in FIG. 3 are input to the DNN. The DNN can describe a relationship between an input acoustic feature $x_0 \in R^{n_0}$ and a modeling unit j in an output layer according to the following function mapping:

$$z_i = x_{i-1} W_i^T + B_i, 1 \leq i \leq N + 1; \text{(formula I)}$$

$$x_i = \sigma(z_i), 1 \leq i \leq N; \text{(formula II)}$$

$$y_j = \frac{\exp(z_{N+1,j})}{\sum_k \exp(z_{N+1,k})}; \text{ (formula III)}$$

where $x_{i,i>0} \in R^{n_i}$ is an output of a hidden layer, $W_i \in R^{n_i \times n_{i-1}}$ and $B_i \in R^{n_i}$ are weights and offset parameters respectively, n_i is the number of nodes on the i^{th} layer, $\theta = \{W_i, B_i\}$, " T " denotes transpose of a matrix, N is the number of the hidden layers, and σ is a nonlinear activation function, for example, a Rectified Linear Unit (ReLU) function $\sigma(z) = \max(z, 0)$. The formula III is a softmax function, representing estimated posterior of a label unit j .

In an actual application, a Recurrent Neural Network (RNN) can also be used in conjunction with the CTC. However, in the case where the training data is restricted such as in the Background, the minimum computing and power consumption requirements of mobile devices can be more easily met by using the DNN in conjunction with the CTC. In order to reduce the complexity in computation, the DNN with about hundreds of nodes in a hidden layer is more suitable.

FIG. 6 is a schematic structural diagram of the CTC in FIG. 3 in an actual application scenario according to an embodiment of this specification.

The CTC is specifically designed for sequential label tasks. Unlike the cross-entropy criterion for the frame-level alignment between input features and target labels, the CTC is aimed to automatically learn the alignment between speech data and label sequences (e.g., phonemes or characters, etc.), thereby eliminating the need for forced alignment of data, and the input is not necessarily the same as the label length.

In FIG. 6, a specified modeling unit is extracted from L , and the CTC is located on a softmax layer of the DNN. The DNN is composed of an $|L|$ unit and a blank unit. The introduction of the blank unit alleviates the burden of label prediction, as symbols are not output correspondingly during uncertainty.

$y_j^t (j \in [0, |L|], t \in [0, T))$ is defined as a probability that the DNN outputs j in a time step t . An input sequence x^T of a frame length T and a target label l^T are given, and $l_i \in L$. A CTC path $\pi=(\pi_0, ..., \pi_{T-1})$ is a frame-level label sequence, which is different from l in that the CTC path allows the appearance of repeated non-blank labels and blank units.

5 The CTC path π can be mapped to its corresponding label sequence l by deleting the repeated labels and the blank units. For example, $\tau("aa-b-c") = \tau("abb-cc-") = "abc"$. A many-to-one mapping function is defined as τ , and "-" represents blank. If x^T is given, and an output probability condition of each time step is assumed to be independent, the probability of the path π is:

10
$$p(\pi|x; \theta) = \prod_{t=0}^{T-1} y_{\pi_t}^t; \text{ (formula IV)}$$

Then, the probability of l can be calculated based on τ by summing the probabilities of all the paths mapped to l . In actual applications, it is troublesome to sum up all the paths on the CTC in terms of calculation. With respect to this problem, a forward and backward dynamic programming algorithm can be adopted. All possible CTC paths are represented
15 compactly as grids based on the algorithm, just as shown in FIG. 6.

At the time of training, the CTC is aimed to enable $\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{(x,l) \in S} -\log(p(l|x; \theta))$, where S represents training data used. At the time of decoding, when the confidence score output by the CTC is greater than a set threshold, a detection engine can make a positive decision accordingly, and it can be considered that
20 corresponding keywords have been detected. The set threshold can be fine-tuned based on a verification data set.

In the embodiment of this specification, the model can be trained by a gradient descent method, preferably by an asynchronous random gradient descent method, to iteratively optimize parameters in the speech wakeup model until the training converges.

For example, the DNN and the CTC can be trained on a server having a Graphics Processing Unit (GPU). Network parameters are randomly initialized to be uniformly distributed within a range of $(-0.02, 0.02)$, an initial learning rate is 0.008, and a momentum is 0.9.

5 For ease of understanding, the learning rate and its function are described as follows.

The learning rate is a parameter used in the gradient descent method. In the gradient descent method, a solution may be initialized first, and on the basis of this solution, a moving direction and a moving step size are determined, so that after the initial solution moves according to the direction and the step size, the output of a target function can be reduced.

10 Then it is updated to a new solution, a next moving direction and a next step size are searched continuously, and after this process is performed iteratively, the target function is constantly decreased, to finally find a solution, such that the target function is relatively small. In the process of finding the solution, if the step size is too large and the search is not careful enough, a good solution may be bypassed, and if the step size is too small, the process of
15 finding the solution will proceed too slowly. Therefore, it is important to set the step size appropriately. The learning rate is used for the adjustment of the original step size. In the gradient descent method, the step size in each adjustment is equal to the learning rate multiplied by a gradient.

In the foregoing training, a verification data set may also be used to cross-verify the
20 speech wakeup model to determine whether the training converges.

In order to improve the performance and robustness of the speech wakeup model, more measures are further provided in the solutions of this specification.

One measure is adaptive training. Specifically, a general model can be fine-tuned with speech data of some specific keywords and at a relatively low learning rate. Based on this
25 consideration, when the speech wakeup model is trained, keyword-specific speech data can

also be acquired, and the speech wakeup model can be trained with the keyword-specific speech data. A learning rate used in the training is less than that used in the training of the speech wakeup model with the general speech data.

Another measure is transfer learning. Specifically, the network parameters may not be randomly initialized, but refer to an existing corresponding network which has the same topology structure as the target network except for fine-grained units in the output layer, and may use a cross entropy criterion. The transfer learning can be considered especially when the training data has a large scale.

Other measures include, for example, using architecture-related vector instructions (e.g., ARM's NEON) to further accelerate multiplication, and so on.

As mentioned previously, the solution of this specification can easily support user-defined speech wakeup keywords. A target label sequence corresponding to such user-defined keywords can be determined through a dictionary.

A speech wakeup method provided in the embodiment of this specification is as described in the foregoing. Based on the same specification idea, a corresponding apparatus is further provided in an embodiment of this specification, as shown in FIG. 7.

FIG. 7 is a schematic structural diagram of a speech wakeup apparatus corresponding to FIG. 2 according to an embodiment of this specification. The dashed boxes represent optional modules. The apparatus may be located on the executive body of the flow in FIG. 2, including an input module 701 and a speech wakeup model 702.

Speech data is input by the input module 701 to the speech wakeup model 702 trained with general speech data, and the speech wakeup model 702 outputs a result for determining whether to execute speech wakeup, wherein the speech wakeup model includes a DNN and a CTC.

Optionally, the general speech data includes a LVCSR corpus.

Optionally, the apparatus further includes a training module 703; and

training, by the training module 703, the speech wakeup model with the general speech data includes:

iteratively optimizing, by the training module 703, parameters in the speech wakeup
5 model with the general speech data by an asynchronous stochastic gradient descent method until the training converges.

Optionally, the training module 703 further acquires keyword-specific speech data; and
trains the speech wakeup model with the keyword-specific speech data, wherein a
learning rate used in the training is less than that used in the training of the speech wakeup
10 model with the general speech data.

Optionally, the training module 703 cross-verifies the speech wakeup model with a verification data set in the training to determine whether the training converges.

Optionally, the outputting, by the speech wakeup model 702, a result for determining whether to execute speech wakeup specifically includes:

15 extracting, by the speech wakeup model 702, acoustic features from the input speech data;

inputting the acoustic features to the DNN included in the speech wakeup model 702 for processing to obtain a class probability of the acoustic features respectively corresponding to each pronunciation phoneme;

20 inputting the class probability to the CTC included in the speech wakeup model 702 for processing to obtain a confidence score of a speech wakeup term corresponding to a pronunciation phoneme sequence; and

determining whether to execute wakeup according to the confidence score, and outputting a determination result.

Optionally, the extracting, by the speech wakeup model 702, acoustic features from the input speech data specifically includes:

- extracting, by the speech wakeup model 702, acoustic feature frames of the input speech data from a window according to a specified time interval, wherein each of the acoustic feature frames is multi-dimension log filter bank energies;
- stacking a plurality of adjacent acoustic feature frames respectively; and
- taking the stacked acoustic feature frames respectively as acoustic features extracted from the monitored speech.

Based on the same specification idea, a corresponding electronic device is further provided in an embodiment of this specification, including:

- at least one processor; and
- a memory communicatively connected to the at least one processor; wherein the memory stores an instruction executable by the at least one processor, and the instruction is executed by the at least one processor to enable the at least one processor to:
 - input speech data to a speech wakeup model trained with general speech data, and output, by the speech wakeup model, a result for determining whether to execute speech wakeup, wherein the speech wakeup model includes a DNN and a CTC.

Based on the same specification idea, a corresponding non-volatile computer storage medium with a computer executable instruction stored thereon is further provided in an embodiment of this specification, wherein the computer executable instruction is configured to:

- input speech data to a speech wakeup model trained with general speech data, and output, by the speech wakeup model, a result for determining whether to execute speech wakeup, wherein the speech wakeup model includes a DNN and a CTC.

Specific embodiments of this specification have been described in the foregoing. Other embodiments fall within the scope of the appended claims. Under some circumstances, the actions or steps described in the claims may be performed in a sequence different from that in the embodiments and still can achieve a desired result. In addition, the processes depicted
5 in the accompanying drawings are not necessarily required to achieve the desired result according to the specific sequence or consecutive sequence shown. Multitask processing and parallel processing are also possible or may be advantageous in some implementation manners.

The embodiments in the specification are described progressively, identical or similar
10 parts of the embodiments may be obtained with reference to each other, and each embodiment emphasizes a part different from other embodiments. Especially, the apparatus, electronic device, and non-volatile computer storage medium embodiments are basically similar to the method embodiments, thus being described simply. For related parts, refer to the descriptions of the parts in the method embodiments.

15 The apparatus, the electronic device, and the non-volatile computer storage medium provided in the embodiments of this specification are corresponding to the method. Therefore, the apparatus, the electronic device, and the non-volatile computer storage medium also have beneficial technical effects similar to those of the corresponding method. As the beneficial technical effects of the method have been described in detail in the
20 foregoing, the beneficial technical effects of the apparatus, the electronic device, and the non-volatile computer storage medium will not be elaborated here.

In the 1990s, an improvement on a technology can be obviously distinguished as an improvement on hardware (e.g., an improvement on a circuit structure such as a diode, a transistor, and a switch) or an improvement on software (an improvement on a method
25 procedure). However, with the development of technologies, improvements of many method

procedures at present can be considered as direct improvements on hardware circuit structures. Almost all designers program the improved method procedures into hardware circuits to obtain corresponding hardware circuit structures. Therefore, it is improper to assume that the improvement of a method procedure cannot be implemented by using a hardware entity module. For example, a Programmable Logic Device (PLD) (e.g., a Field Programmable Gate Array (FPGA)) is such an integrated circuit, and its logic functions are determined by a user programming device. Designers program by themselves to "integrate" a digital system into a PLD, without asking a chip manufacturer to design and manufacture a dedicated integrated circuit chip. Moreover, at present, the programming is mostly implemented by using logic compiler software instead of manually manufacturing an integrated circuit chip. The logic compiler software is similar to a software compiler used for developing and writing a program, and original codes before compiling also need to be written by using a specific programming language, which is referred to as a Hardware Description Language (HDL). There are many types of HDLs, such as Advanced Boolean Expression Language (ABEL), Altera Hardware Description Language (AHDL), Confluence, Cornell University Programming Language (CUPL), HDCal, Java Hardware Description Language (JHDL), Lava, Lola, MyHDL, PALASM, and Ruby Hardware Description Language (RHDL), among which Very-High-Speed Integrated Circuit Hardware Description Language (VHDL) and Verilog are most commonly used now. Those skilled in the art should also know that a hardware circuit for implementing the logic method procedure can be easily obtained by slightly logically programming the method procedure using the above several hardware description languages and programming it into an integrated circuit.

A controller can be implemented in any suitable manner. For example, the controller can employ a form of a microprocessor or a processor and a computer readable medium that

stores computer readable program codes (such as software or firmware) executable by the microprocessor or processor, a logic gate, a switch, an Application Specific Integrated Circuit (ASIC), a programmable logic controller, and an embedded microcontroller. Examples of the controller include, but are not limited to, the following microcontrollers:

5 ARC 625D, Atmel AT91SAM, Microchip PIC18F26K20, and Silicone Labs C8051F320.

The controller of the memory can further be implemented as a part of control logic of the memory. Those skilled in the art also know that in addition to implementing the controller by using pure computer readable program codes, it is completely feasible to logically program the method steps to enable the controller to implement the same function in a form
10 of a logic gate, a switch, an ASIC, a programmable logic controller, and an embedded microcontroller. Therefore, such a controller may be considered as a hardware component, and apparatuses included in the controller and configured to implement various functions may also be considered as structures inside the hardware component. Or, the apparatuses configured to implement various functions may even be considered as both software
15 modules configured to implement the method and structures inside the hardware component.

The system, apparatus, modules or units illustrated in the foregoing embodiments specifically can be implemented by a computer chip or an entity, or implemented by a product having a specific function. A typical implementation device is a computer. Specifically, for example, the computer can be a personal computer, a laptop computer, a
20 cellular phone, a camera phone, a smart phone, a personal digital assistant, a media player, a navigation device, an email device, a game console, a tablet computer, a wearable device, or a combination of any of these devices.

For ease of description, the apparatus is divided into various units based on functions, and the units are described separately. In an implementation of this specification, the

functions of various units can also be implemented in one or more pieces of software and/or hardware.

Those skilled in the art should understand that the embodiments of this specification can be provided as a method, a system, or a computer program product. Therefore, the
5 embodiments of this specification may be implemented in a form of a complete hardware embodiment, a complete software embodiment, or an embodiment combining software and hardware. Moreover, the embodiments of this specification can be in the form of a computer program product implemented on one or more computer usable storage media (including, but not limited to, a magnetic disk memory, a CD-ROM, an optical memory and the like)
10 including computer usable program codes.

This specification is described with reference to flowcharts and/or block diagrams of the method, the device (system) and the computer program product according to the embodiments of this specification. It should be understood that computer program instructions may be used to implement each process and/or block in the flowcharts and/or
15 block diagrams and combinations of processes and/or blocks in the flowcharts and/or block diagrams. The computer program instructions may be provided to a general-purpose computer, a special-purpose computer, an embedded processor or a processor of another programmable data processing device to generate a machine, such that the computer or the processor of another programmable data processing device executes an instruction to
20 generate an apparatus configured to implement functions designated in one or more processes in a flowchart and/or one or more blocks in a block diagram.

The computer program instructions may also be stored in a computer readable memory that can guide the computer or another programmable data processing device to work in a specific manner, such that the instruction stored in the computer readable memory generates
25 an article of manufacture including an instruction apparatus, and the instruction apparatus

implements functions designated by one or more processes in a flowchart and/or one or more blocks in a block diagram.

The computer program instructions may also be loaded to the computer or another programmable data processing device, such that a series of operation steps are executed on
5 the computer or another programmable device to generate a computer implemented processing, and therefore, the instruction executed in the computer or another programmable device provides steps for implementing functions designated in one or more processes in a flowchart and/or one or more blocks in a block diagram.

In a typical configuration, the computing device includes one or more central processing
10 units (CPUs), an input/output interface, a network interface, and a memory.

The memory can include computer readable media such as a volatile memory, a Random Access Memory (RAM), and/or non-volatile memory, e.g., a Read-Only Memory (ROM) or a flash RAM. The memory is an example of a computer readable medium.

The computer readable medium includes non-volatile and volatile media as well as
15 movable and non-movable media, and can implement information storage by means of any method or technology. The information can be a computer readable instruction, a data structure, and a module of a program or other data. An example of the storage medium of a computer includes, but is not limited to, a phase change memory (PRAM), a static random access memory (SRAM), a dynamic random access memory (DRAM), other types of RAM,
20 a ROM, an electrically erasable programmable read-only memory (EEPROM), a flash memory or other memory technologies, a compact disk read-only memory (CD-ROM), a digital versatile disc (DVD) or other optical storages, a cassette tape, a magnetic tape/magnetic disk storage or other magnetic storage devices, or any other non-transmission medium, and can be used to store information accessible to the computing device. According

to the definition in this text, the computer readable medium does not include transitory media, such as a modulated data signal and a carrier.

It should be further noted that the terms "include", "comprise" or any other variations thereof are intended to cover non-exclusive inclusion, so that a process, method, commodity or device including a series of elements not only includes the elements, but also includes other elements not expressly listed, or further includes elements inherent to the process, method, commodity or device. In the absence of more limitations, an element defined by "including a/an..." does not exclude that the process, method, commodity or device including the element further has other identical elements.

This specification can be described in a general context of a computer executable instruction executed by a computer, for example, a program module. Generally, the program module includes a routine, a program, an object, an assembly, a data structure, and the like used for executing a specific task or implementing a specific abstract data type. This specification can also be implemented in distributed computing environments. In these distributed computing environments, a task is executed by using remote processing devices connected via a communications network. In the distributed computing environments, the program module may be located in local and remote computer storage media including a storage device.

The embodiments in the specification are described progressively, identical or similar parts of the embodiments may be obtained with reference to each other, and each embodiment emphasizes a part different from other embodiments. Especially, the system embodiment is basically similar to the method embodiments, thus being described simply. For related parts, refer to the descriptions of the parts in the method embodiments.

The foregoing description is merely embodiments of this specification, and is not used to limit this application. For those skilled in the art, this application can have various

alterations and changes. Any modification, equivalent replacement, improvement and the like made without departing from the spirit and principle of this application should be included in the scope of the claims of this application.

CLAIMS

1. A speech wakeup method, comprising:

inputting speech data to a speech wakeup model trained with general speech data, and
outputting, by the speech wakeup model, a result for determining whether to execute speech
5 wakeup, wherein the speech wakeup model comprises a Deep Neural Network (DNN) and
a Connectionist Temporal Classifier (CTC).

2. The method of claim 1, wherein the general speech data comprises a Large
Vocabulary Continuous Speech Recognition (LVCSR) corpus.

3. The method of claim 1, wherein the training the speech wakeup model with the
10 general speech data comprises:

iteratively optimizing parameters in the speech wakeup model with the general speech
data by means of an asynchronous stochastic gradient descent method until the training
converges.

4. The method of claim 3, further comprising:

15 acquiring keyword-specific speech data; and

training the speech wakeup model with the keyword-specific speech data, wherein a
learning rate used in the training is less than that used in the training of the speech wakeup
model with the general speech data.

5. The method of claim 3, further comprising:

20 cross-verifying the speech wakeup model with a verification data set in the training to
determine whether the training converges.

6. The method of claim 1, wherein the outputting, by the speech wakeup model, a result
for determining whether to execute speech wakeup specifically comprises:

extracting acoustic features from the input speech data;

inputting the acoustic features to the DNN comprised in the speech wakeup model for processing to obtain a class probability of the acoustic features respectively corresponding to each pronunciation phoneme;

inputting the class probability to the CTC comprised in the speech wakeup model for processing to obtain a confidence score of a speech wakeup term corresponding to a pronunciation phoneme sequence; and

determining whether to execute wakeup according to the confidence score, and outputting a determination result.

7. The method of claim 6, wherein the extracting acoustic features from the input speech data specifically comprises:

extracting acoustic feature frames of the input speech data from a window according to a specified time interval, wherein each of the acoustic feature frames is multi-dimension log filter bank energies;

stacking a plurality of adjacent acoustic feature frames respectively; and

taking the stacked acoustic feature frames respectively as acoustic features extracted from the monitored speech.

8. A speech wakeup apparatus, comprising an input module and a speech wakeup model;

wherein speech data is input by the input module to the speech wakeup model trained with general speech data, and the speech wakeup model outputs a result for determining whether to execute speech wakeup, wherein the speech wakeup model comprises a Deep Neural Network (DNN) and a Connectionist Temporal Classifier (CTC).

9. The apparatus of claim 8, wherein the general speech data comprises a large vocabulary continuous speech recognition (LVCSR) corpus.

10. The apparatus of claim 8, further comprising a training module;

wherein training, by the training module, the speech wakeup model with the general speech data comprises:

iteratively optimizing, by the training module, parameters in the speech wakeup model with the general speech data by means of an asynchronous stochastic gradient descent
5 method until the training converges.

11. The apparatus of claim 10, wherein the training module further acquires keyword-specific speech data; and

trains the speech wakeup model with the keyword-specific speech data, wherein a learning rate used in the training is less than that used in the training of the speech wakeup
10 model with the general speech data.

12. The apparatus of claim 10, wherein the training module cross-verifies the speech wakeup model with a verification data set in the training to determine whether the training converges.

13. The apparatus of claim 8, wherein the outputting, by the speech wakeup model, a
15 result for determining whether to execute speech wakeup specifically comprises:

extracting, by the speech wakeup model, acoustic features from the input speech data;

inputting the acoustic features to the DNN comprised in the speech wakeup model for processing to obtain a class probability of the acoustic features respectively corresponding to each pronunciation phoneme;

20 inputting the class probability to the CTC comprised in the speech wakeup model for processing to obtain a confidence score of a speech wakeup term corresponding to a pronunciation phoneme sequence; and

determining whether to execute wakeup according to the confidence score, and outputting a determination result.

14. The apparatus of claim 6, wherein the extracting, by the speech wakeup model, acoustic features from the input speech data specifically comprises:

extracting, by the speech wakeup model, acoustic feature frames of the input speech data from a window according to a specified time interval, wherein each of the acoustic
5 feature frames is multi-dimension log filter bank energies;
stacking a plurality of adjacent acoustic feature frames respectively; and
taking the stacked acoustic feature frames respectively as acoustic features extracted from the monitored speech.

15. An electronic device, comprising:

10 at least one processor; and
a memory communicatively connected to the at least one processor; wherein
the memory stores an instruction executable by the at least one processor, and the instruction is executed by the at least one processor to enable the at least one processor to:
input speech data to a speech wakeup model trained with general speech data, and
15 output, by the speech wakeup model, a result for determining whether to execute speech wakeup, wherein the speech wakeup model comprises a Deep Neural Network (DNN) and a Connectionist Temporal Classifier (CTC).