TEXT INPUTTING METHOD, APPARATUS AND SYSTEM

FIELD OF THE DISCLOSURE

[0001] The present disclosure relates to text inputting techniques, and more particularly, to a text inputting method, apparatus and system.

BACKGROUND OF THE DISCLOSURE

[0002] Input method software is a kind of frequently used word inputting system. ~~A general operation procedure of the input method software is as follows: the input method software receives a code sequence (such as Pinyin or five strokes) inputted by a user via a keyboard, finds a candidate word list corresponding to the code sequence using a universal language model by taking the code sequence as a parameter. The input method software calculates an on-screen probability of each candidate word in the candidate sentence list, sorts the candidate words according to their on-screen probabilities and then displays the candidate word list to the user. The user finishes the input through selecting a required word from the candidate word list.~~

[0003] A traditional text inputting method usually uses a universal language model as a core to construct the input method. The universal language model is obtained through analyzing a large amount of training material. The training material is usually obtained automatically from the Internet. It represents general an input requirement of most users, i.e., the universal language model is created based on common word-selection habits of most users. However, when inputting words using the input method software, a user usually wants to obtain characters that he constantly uses rapidly. When selecting a word, different users may desire different kinds of candidate sentences ranking in the first according to their different identities, interests and habits, and field of characters that they used to use. For example, a scientific researcher and a bank clerk usually hope that professional terms in their fields are ranked in the first when inputting words. For another example, people from north-east and Sichuan province also hope their own dialect words ranking in the first in the candidate sentence list. The traditional input method which uses the universal language model cannot meet the input requirement of

be stored in the server or in a user device.

[0032] The user language model is obtained through training according to the inputs of the user. It should be noted that, during a first input using the input method software, since the user language model has not been updated, the on-screen probability of each candidate sentence in the candidate sentence list may be calculated according to the universal language model. The method and principle are similar to those of conventional input method using the universal language model and is not described herein.

[0033] Each time the user inputs text using the input method software, words inputted by the user are recorded. The user language model is updated according to information of the words inputted by the user. The user language model is stored in association with the user identifier. When next time a word is inputted, the on-screen probability of each candidate sentence in the candidate work list can be calculated according to both the user language model and the universal language model.

[0034] In one example, both the universal language model and the user language model are stored in the user device. Thus, the universal language model and the user language module are obtained from a local storage of the user device to calculate the on-screen probability of each candidate sentence in the candidate sentence list. In this method, the user device does not need to transmit a request to a server. This manner is referred to as a local input method.

[0035] In another example, the universal language model and the user language model are stored in the server which obtains the universal language model and the user language model to calculate the on-screen probability of each candidate sentence in the candidate sentence list. In this manner, processing of the input method is implemented by the server. This manner is referred to a cloud input method.

[0036] At block S108, the candidate sentences in the candidate sentence list are sorted according to their on-screen probabilities.

[0037] In this example, the candidate sentences in the candidate sentence list are sorted in an ascending order according to their on-screen probabilities. Top-ranked words conform to the user's language habit better and may be desired by the user. Thus, the user may find a required candidate sentence more rapidly, which increase a text input accuracy and a text input speed.

[0001] At block S110, the sorted candidate sentence list is outputted.

[0002] The user may select a required candidate sentence from the candidate sentence

sentence list according to the mixed model.

[0063] The on-screen probability of the candidate sentence in the candidate sentence list is a probability that the candidate sentence may be selected by the user calculated by the mixed model. The greater the on-screen probability, the more forward position that the candidate sentence is ranked in the candidate sentence list. Thus, the user can find a required word more rapidly. The text inputting speed is increased.

[0064] At block S208, the server sorts the candidate sentences in the candidate sentence list according to their on-screen probabilities and transmits the sorted candidate sentence list to the user device.

[0065] At block S210, the user device receives and outputs the sorted candidate sentence list. The user may select a required word from the candidate sentence list. The selected candidate sentence is outputted from the input method software to an application program, e.g., a text document, a notebook, a presentation document, etc.

[0066] In one example, a text inputting method is provided.

[0067] FIG. 3 is a flowchart illustrating a text inputting method according to an example of the present disclosure. As shown in FIG. 3, the method includes the following.

[0068] At block ~~S202~~S302, a client obtains a user identifier and finds a corresponding user language model in the user device according to the user identifier.

[0069] The user identifier identifies the user uniquely. It may be an account registered by the user in the input method software, or an identifier number assigned for the user, or an IP address or MAC address associated with a device of the user. After an identity authentication, the user logs in the input method software. The client obtains the user identifier and transmits the user identifier to the server which finds the corresponding user language model.

[0070] At block ~~S204~~S304, the user device obtains a user input and generates a candidate sentence list according to the user input.

[0071] The user input may be a speech input, handwritten characters, optical characters or strings. The universal language model and the user language model are both stored in the user device. The user language model and the universal language model may be obtained directly from the user device to calculate the on-screen probability of each candidate sentence in the candidate sentence list. In this example, the user device does not need to transmit a request to the server. This method is also referred to as "local input method".

[0072] At block ~~S206~~S306, the user device obtains the universal language model from itself and calculates the on-screen probability of each candidate sentence in the candidate sentence list according to the user language model and the universal language model.

[0073] The universal language model may be a conventional statistical language model obtained through training using a large amount of material obtained from texts inputted by a large number of users on the Internet. The user language model is corresponding to a unique user. Different users correspond to different user language models.

[0074] The user language model is obtained through training according to the inputs of the user. It should be noted that, during the first time input using the input method software, since the user language model has not been updated, the on-screen probability of each candidate sentence in the candidate sentence list may be calculated according to the universal language model. The method and principle are similar to those of conventional input method using the universal language model and is not described herein.

[0075] Each time the user inputs a word using the input method software, the word inputted by the user is recorded. The language model is updated according to the word inputted by the user. The user language model is stored in association with the user identifier. When next time a word is inputted, the on-screen probability of each candidate sentence in the candidate work list can be calculated according to both the user language model and the universal language model.

[0076] In one example, the user language model may use a Bigram language model. The creation of the model has been described above and is not repeated herein.

[0077] In one example, the process of calculating the on-screen probability of each candidate sentence in the candidate sentence list according to the user language model and the universal language model includes: perform a linear interpolation operation to the user language model and the universal language model to generate a mixed model, and calculate the on-screen probability of each candidate sentence in the candidate sentence list according to the mixed model.

[0078] The on-screen probability of the candidate sentence in the candidate sentence list is a probability that the candidate sentence may be selected by the user calculated by the mixed model. The greater the on-screen probability, the more forward position that the candidate sentence is ranked in the candidate sentence list. Thus, the user can find a

required word more rapidly. The text inputting speed is increased.

[0079] At block ~~S208~~S308, the user device sorts the candidate sentences in the candidate sentence list according to their on-screen probabilities and outputs the sorted candidate sentence list.

[0080] In one example, as shown in FIG. 4, a text inputting system includes: a searching module 102, a candidate sentence list generating module 104, a probability calculating module 106, a sorting module 108 and an outputting module 110.

[0081] The searching module 102 obtains a user identifier and finds a corresponding user language model according to the user identifier.

[0082] The user identifier identifies the user uniquely. It may be an account registered by the user in the input method software, or an identifier number assigned for the user, or an IP address or MAC address associated with a device of the user.

[0083] In one example, as shown in FIG. 5, the above text inputting system may further include: a user language model creating module 112 and a user language model updating module 114.

[0084] The user language model creating module 112 may be configured in the user device or at the server side. The created user language model may be stored in the user device or in the server.

[0085] The user language model updating module 114 updates, each time the user inputs a word, the user language model according to the word inputted by the user.

[0086] The user language model updating module 114 may be located in the user device or in the server side. The updated user language model may be stored in the user device or uploaded by the user device to the server for storage. Thus, the user language model in the server may be updated continuously along with the input of the user. Thus, the user language model in the server becomes more and more accurate. When the user uses the input method software on different user devices, the server issues the newest user language model to the user device. Thus, the synchronization of the user language model is realized and it is fit for different user devices.

[0087] The candidate sentence list generating module 114 obtains the user input and generates a candidate sentence list according to the user input.

[0088] The user input may be a speech input, handwritten characters, optical characters or strings. A conventional text inputting method may be used to find candidate sentences that match the user input to generate the candidate sentence list.

[0089] In one example, the candidate sentence list generating module 104 may be in

GONG" previously. After recognizing based on the standard Ngram language model, three characters "XIAO", "JING" and "HUI", and words "SHI", "TENG XUN YUAN GONG" are cached in the cache area. When the user inputs "XIAO JING HUI XIE LE YI PIAN ZHUAN LI" later, the three characters "XIAO", "JING" and "HUI" stored in the cache area in the cache-based language area affect the current inputted sentence of the user. If there is no characters "XIAO", "JING" and "HUI" in the cache area, "XIAO JING HUI" has a higher probability to be falsely converted when the user inputs. However, according to the information cached in the cache area, "XIAO JING HUI" has a higher probability to be correctly converted. Therefore, the input "XIE JING HUI XIE LE YI PIAN ZHUAN LI" has a higher probability to be correctly converted by the input method.

[0166] It can be seen from the above that, the cache-based language model is created continuously based on the inputs of the user. It reflects current input scenario information of the user, and reflects inputting habits of the user on the other hand. The combining of the standard Ngram language model with the cache-based language model effectively learns from and is self-adapted to the input scenario and inputting habit of the user.

[0167] At block 707, an output sentence with a largest probability is outputted.

[0168] At block 708, the sentence cached in the cache area is updated according to the outputted sentence.

[0169] Experiments show that, compared with the standard Ngram language model, the cache-based language modeling method provided by the example of the present disclosure has a higher accuracy ratio for recognizing the input of the user. In addition, the Chinese input software based on this modeling method is more intelligent.

[0170] It should be noted that, the language modeling method provided by the example of the present disclosure is applicable not only for Chinese input method, but also input methods of other Asian languages such as Japanese, Korean, and Cambodia. The modeling method for these languages is similar to that of Chinese and is not repeated herein.

[0171] FIG. 8 is a schematic diagram illustrating a structure of a language modeling apparatus according to an example of the present disclosure. As shown in FIG. 8, the apparatus includes: a standard Ngram language model module 801, a cache module 802, a cache-based language modeling module 803 and a mixed model module 804.

[0172] The standard Ngram language model module 801 receives an input of the user,

respectively calculates a standard conditional probability of each word in the user input and outputs the conditional probabilities to the mixed model module 804.

[0173] In this example, the formula for calculating the conditional probability of each word by the standard Ngram language model module 801 is as follows.

$$p(w_i / w_{i-n+1}w_{i-n+2}\dots w_{i-1}) = \frac{C(w_{i-n+1}\dots w_{i-1}w_i)}{C(w_{i-n+1}\dots w_{i-2}w_{i-1})},$$

wherein $C(w_{i-n+1}\dots w_{i-1}w_i)$ denotes a number of times that a word sequence $w_{i-n+1}\dots w_{i-1}w_i$ emerges in training material of the standard Ngram language model;

w_i denotes the i th word in the Chinese sentence; and

n is a preconfigured constant.

[0174] The cache module 802 caches a sentence outputted by the mixed model module 804.

[0175] The cache-based language modeling module 803 respectively calculates a conditional probability of each word in the user input according to the user input and sentences cached in the cache module 802 based on a pre-configured cache-based language modeling policy, and outputs the conditional probabilities to the mixed model module 804.

[0176] In this example, the cache-based language modeling module 803 may calculate the conditional probability of each word according to a following formula.

$$p_{cache}(w_i / w_{i-n+1}w_{i-n+2}\dots w_{i-1}, t_i) = f(t_i)x\frac{c(w_{i-n+1}\dots w_{i-1}w_i)}{c(w_{i-n+1}\dots w_{i-2}w_{i-1})},$$

wherein $C(w_{i-n+1}\dots w_{i-1}w_i)$ denotes a number of times that the word sequence $w_{i-n+1}\dots w_{i-1}w_i$ emerges in training material of the cache-based language model;

w_i denotes the i th word in the Chinese sentence;

n is a preconfigured constant; and

$f(t_i)$ denotes a time function.

[0177] The mixed model module 804 calculates a mixed conditional probability of each word according to the standard conditional probability and the cache conditional probability, obtains a sentence probability of each output sentence according to the mixed conditional probability, selects and outputs an output sentence with a largest probability.

[0178] In one example, a calculation formula for the mixed conditional probability is as follows.

$$p_{mixture}(w_i / w_{i-n+1}w_{i-n+2}\ldots w_{i-1}) =$$
$$\alpha xp(w_i / w_{i-n+1}w_{i-n+2}\ldots w_{i-1}) + (1-\alpha)xp'_{cache}(w_i / w_{i-n+1}w_{i-n+2}\ldots w_{i-1}, t_i)$$

wherein is an interpolation parameter which is a constant between 0 and 1.

[0179] The sentence probability of the output sentence may be calculated as follows.

$$P(S) = \prod_{i=1}^{m} p_{mixture}(w_i / w_{i-n+1}w_{i-n+2}\ldots w_{i-1}),$$

wherein m denotes the number of words included in the Chinese sentence.

[0180] The standard Ngram language model module 801 includes: a first word sequence frequency counting unit, a second word sequence frequency counting unit and a standard conditional probability calculating unit (not shown in the Figure).

[0181] The first word sequence frequency counting unit obtains a number of times k'_i that a word sequence which includes the ith word and a predefined constant number of words before the ith word emerges in the training material of the standard Ngram language model, and outputs k'_i to the standard conditional probability calculating unit.

[0182] The second word sequence frequency counting unit obtains a number of times k'_{i-1} that a word sequence which includes the predefined constant number of words before the ith word emerges in the training material of the standard Ngram language model, and outputs k'_{i-1} to the standard conditional probability calculating unit.

[0183] The standard probability calculating unit calculates a ratio of k'_i to k'_{i-1}, takes the ratio as the standard conditional probability of the ith word in the user input.

[0184] The cache-based language modeling module 803 includes: a third word sequence frequency counting unit, a fourth word sequence frequency counting unit, a time function value obtaining unit and a cache conditional probability calculating unit (not shown in the Figure).

[0185] The third word sequence frequency counting unit obtains a number of times k_i that a word sequence which includes the ith word and a predefined constant number of words before the ith word emerges in the cached training material, and outputs k_i to the standard conditional probability calculating unit.

[0186] The fourth word sequence frequency counting unit obtains a number of times k_{i-1} that a word sequence which includes the predefined constant number of words before the ith word emerges in the cached training material, and outputs k_{i-1} to the standard conditional probability calculating unit.

[0187] The time function value calculating unit obtains a time function value of the i th word and outputs the value to the cache conditional probability calculating unit.

[0188] The cache conditional probability calculating unit calculates a ratio of k_i to k_{i-1}, multiplies the ratio with the time function value of the i th word to obtain the cache conditional probability of the i th word in the user input.

[0189] The mixed model module 804 includes: an interpolation parameter storage unit, a first multiplying unit, a second multiplying unit, a mixed conditional probability calculating unit, a sentence probability calculating unit and an output sentence selecting unit (not shown in the Figure).

[0190] The interpolation parameter storage unit stores an interpolation parameter whose value is configured between 0 and 1 in advance.

[0191] The first multiplying unit calculates a product of the interpolation parameter stored in the interpolation parameter storage unit and the standard conditional probability of the i th word and outputs the product to the mixed conditional probability calculating unit.

[0192] The second multiplying unit calculates a difference between 1 and the interpolation and calculates a product of the different and the cache conditional probability of the i th word, and outputs the product to the mixed conditional probability calculating unit.

[0193] The mixed conditional probability calculating unit adds the products related to the i th word and takes the sum as the mixed conditional probability of the i th word.

[0194] The sentence probability calculating unit calculates a product of the mixed conditional probability of each word to obtain the sentence probability of an output sentence.

[0195] The output sentence selecting unit selects a maximum sentence probability calculated by the sentence probability calculating unit and outputs the output sentence corresponding to the maximum sentence probability.

[0196] It can be seen from the above that, according to the language modeling method and apparatus provided by the examples of the present disclosure, the user input is cached. The cached user input is relevant to user input history information and a user input scenario. Thus, the cached-based language model has a self-study function which makes the language model more intelligent. On the other hand, through learning each inputting habit of the user, the man-machine interaction software is capable of being

the user language model according to information of the word inputted by the user.

25. The text inputting system of claim ~~14~~24, wherein the user language model updating module is further to record word information and word frequency information of the word inputted by the user, obtain the word information and the word frequency information, dividing the word, perform a frequency processing to the divided word according to the word frequency information, and update the user language model according to the divided word and the processed word frequency information.

26. The text inputting system of any one of claims 23 to 25, wherein the on-screen probability generating module is further to perform a linear interpolation to the user language model and the universal language model to generate a mixed model, calculate the on-screen probability of each candidate sentence in the candidate sentence list according to the mixed model.

27. A text processing system, comprising: a client and a server; wherein

the client is to obtain a user identifier and find a user language model corresponding to the user identifier according to the user identifier, obtain a user input, upload the user input to the server, receive a sorted candidate sentence list from the server and output the sorted candidate sentence list;

the server is to generate a candidate sentence list according to the user input, obtain a universal language model, calculate an on-screen probability of each candidate sentence in the candidate sentence list according to the user language model and the universal language model, sort the candidate sentences in the candidate sentence list according to their on-screen probabilities, and issue the sorted candidate sentence list to the client.

28. The text processing system of claim 27, wherein the server is further to perform a linear interpolation to the user language model and the universal language model to generate a mixed model, and calculate the on-screen probability of each candidate sentence in the candidate sentence list according to the mixed model.

29. The text processing system of claim 27, further comprising:

a character processing unit; wherein