

Speech Generation from Concept for Realizing Conversation with an Agent in a Virtual Room

Keikichi Hirose, Junji Tago** and Nobuaki Minematsu****

*Graduate School of Frontier Sciences, **Graduate School of Engineering, ***Graduate School of Information Science and Technology

University of Tokyo, Japan

{hirose, tago, mine}@gavo.t.u-tokyo.ac.jp

Abstract

A concept to speech generation was realized in an agent dialogue system, where an agent (a stuffed animal) walked around in a small room constructed on a computer display to complete some jobs with instructions from a user. The communication between the user and the agent was done through speech. If the agent could not complete the job because of some difficulties, it tried to solve the problems through conversations with the user. Different from other spoken dialogue systems, the speech output from the agent was generated directly from the concept, and was synthesized using higher linguistic information. This scheme could largely improve the prosodic quality of speech output. In order to realize the concept to speech conversion, the linguistic information was handled as a tree structure in the whole dialogue process.

1. Introduction

Because of recent advancement in speech recognition technology, a number of spoken dialogue systems have been developed. However, research works on speech output generation are rather few, and in most systems, text-to-speech (TTS) conversion devices are used for the purpose. During reply sentence generation, the system may have rich information, such as important words, syntactic structure of the sentence and so on, which should be reflected on prosody of reply speech. However, this process is rather difficult when we utilize commercially available TTS devices. Moreover, misreading may occur because of wrong linguistic processing in the TTS devices. From this viewpoint, we have been trying to realize a scheme to directly converting concept of system reply into speech.

In our previous system to retrieve academic documents by voice, we realized a concept-to-speech conversion scheme, where the syntactic structure and focal positions of generated sentences were utilized to control prosodic features [1]. However, the method is based on selecting one of pre-defined concepts, for each of which a sentence template was prepared. A focus on the study was much more to realize output speech easy to be understood by users by placing emphasis on important words [2].

In the current paper, with the aim of realizing a more generalized scheme of concept-to-speech (CTS) conversion, a spoken dialogue system was constructed where an agent in a virtual room on computer display walks about in the room and do a job, such as to move the vase, following to the user's instruction. When the agent found some difficulties, it asked the user for help. Although, the conversation between the

agent and the user is limited to a simple one in the current system, it can be a more complex one, which will require a sophisticated CTS conversion scheme. The system handles dynamically changing information, which is usually not the case for information retrieving systems, and, therefore, misunderstanding between the user and the agent may frequently occur. This requires a variety of reply sentences. Several similar agent systems were already developed, but the conversation was done by typing keyboard and generating sentences on display without spoken dialogue [3].

The rest of the paper is constructed as follows: In section 2, the developed system is outlined. Then, in section 3, explanation is given on how the linguistic information is handled in the system. Dialogue processing and sentence generation of the system are explained in sections 4 and 5, respectively. An example of dialogue is given in section 6. Section 7 concludes the paper.

2. System outline

As shown in Fig. 1, the developed system consists of four portions for handling speech input and output (speech recognizer, syntax analyzer, dialogue manager, and speech synthesizer), together with two portions for controlling the virtual room constructed on the computer display (virtual space manager and CG generator) [4].

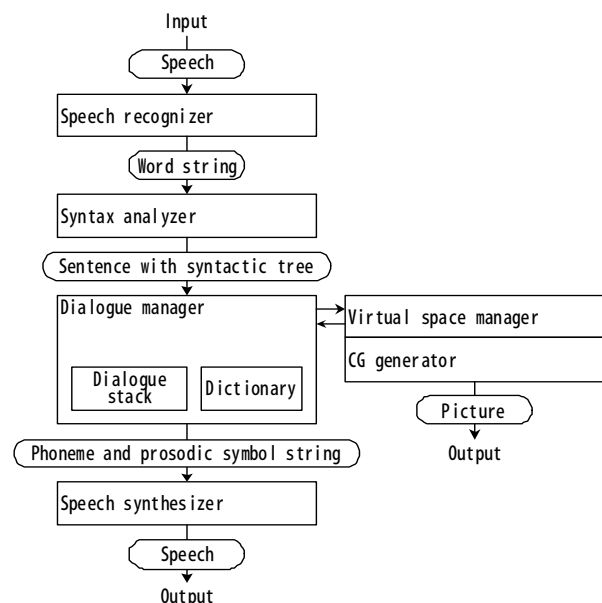


Figure 1: Outline of the agent dialogue system.

The recognizer receives the speech input and converts it into a word string. The grammar-based version of speech recognition software, Julian was used [5]. The syntax analyzer outputs the syntactic structure of the word string through morpheme and syntactic analyses. The morpheme analysis result is obtainable as the output of Julian. As for the syntactic analysis, it is conducted by a simple rules developed by the authors. This is because, in the current system, sentence style and vocabulary are very limited. The dialogue manager first extracts user's intention included in the speech input, and, then, sends the agent the action order suited to the current state of the room. It also generates reply content and converts it into a string of prosodic and phone symbols. The speech synthesizer generates output speech from the string. The synthesis is based on a waveform concatenation with TD-PSOLA prosody modification. The fundamental frequency (F_0) contour generation process model was used to control the F_0 movement [6]. The virtual space manager controls the movement of the agent according to the position of each object in the room. For instance, when the agent cannot move because an object blocks its way, the virtual space manager sends this situation to the dialogue manager. The CG generator enables the real-time display of the room. Figure 2 is an example of picture output showing the virtual room.

Processes in the dialogue manager are explained in the following three sections. The major process of CTS is conducted in the dialogue manager.



Figure 2: An example of picture output. The stuffed bear standing at the center of the room is the agent.

3. Linguistic information

3.1. Dictionary

The dialogue manager has three types of dictionaries, which are necessary to understand user inputs (to get user's intention from the input sentences), and to generate reply sentences. They are part-of-speech dictionary, conjugation dictionary and word dictionary. The part-of-speech dictionary tells us if each word being categorized into a content word or a particle, and indicates how each word concatenates to other words. It has a hierarchical structure; for instance, noun belongs to the upper stage and pronoun belongs to the lower stage. The conjugation dictionary shows the conjugation forms for each

conjugation type. It also keeps information on how each conjugation form is pronounced. The word dictionary stores the following information for each word:

"*identifier*" to specify the word.

"*display*" to transcribe the word. This is the same with identifier in many cases.

"*part*" is the part-of-speech of the word. With this, access to the part-of-speech dictionary is done.

"*stem*" is the word stem.

"*inflection*" is the conjugation type of the word.

"*connection*" to indicate how the word concatenates to other words. If this is null, information in the part-of-speech dictionary is used.

"*phoneme_symbol*" shows the word's pronunciation in phoneme symbols. Information on accent type and accent attribute, which is necessary for speech synthesis, is also included.

"*dialog_data*" is the information for dialogue. It is necessary to understand the user input. (refer to Section 4)

The word dictionary is also used in speech recognizer, syntax analyzer, and speech synthesizer. The vocabulary size is currently around 130.

3.2. Representation of sentence

Input sentences to the dialogue manager are represented in the LISP form so that their syntactic structures are kept throughout the further processes. For instance, the sentence "isuo tsukueno maeni oite (Put the chair in front of the desk.*)" has the syntactic structure shown in Fig. 3, and is written in the LISP form as:

(te(oku(o(isu))(ni(mae(no(tsukue))))))

From now on, this form shall be called word LISP form.

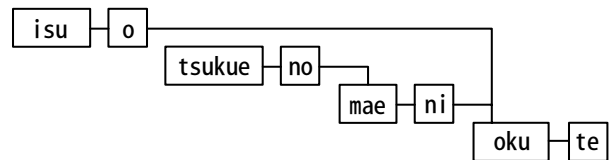


Figure 3: Syntactic structure of the sentence "isuo tsukueno maeni oite (Put the chair in front of the desk.)."

4. Dialogue processing

In the current system, the dialogue is limited to that the user makes a request to the agent in the virtual room to move an object and the agent asks back for help if there are some difficulties. So, the system only handles objects and their positions in the room. The objects are such as red telephone, gray chair, and so on, and shall be called items hereinafter. In the current system, they can be identified only by name and color.

When the dialogue manager receives the user input speech, it activates a dialogue process according to the content of the speech. Although, the current process is limited to the "ordering the agent to move," other processes can easily be added.

The dialogue manager first decides the type of order of the input sentence. Then, it searches in the input sentence for

the phrase with information on the item and the position. When the item and its position are identified, the dialogue manager decides the agent's action. When the system cannot complete these processes, it generates a question to the user to solve the problem. For instance, if the agent's situation is that shown in Fig. 2, it cannot reach the black telephone at the right-back corner responding to such an order, because a vase blocks the path to the telephone. In this case the system generates the sentence: "I cannot move to the telephone. What should I do?" When the problem is solved through dialogue with the user, the agent completes the order included in the first input speech.

When there are more than two items with the same property, the system specifies the one which the user talking about, by tracing the syntactic structure of the user's sentence from the node to the leaf. For instance, the sentence "reizoukono maeno tsukueo motte (Pick up the desk in front of the refrigerator.)" has the structure:

(te(motsu(o(tsukue(no(mae(no(reizouko))))))))

and, in this case, the item "reizouko" is first specified, its position is decided, and then the item "tsukue" is specified. When there is more than one "reizouko," and cannot specify the "tsukue," the system solves the problem through the dialogue with the user.

Table 1: *Examples of State.*

State	Content
movablef_o	Can move to the front of an item.
movablef_p	Can move to the front of a position.
have_nothing	Hands are free.
frontof_o	In front of the item.
have_o	The agent has the item.
nothing_on_p	The position is open.
frontof_p	In front of the position.
o_on_p	The item is at the position.

Table 2: *Examples of Action.*

Action	Initial state	End state
To move to the front of the position.	Possible to move to the front of the place.	In front of the position.
To take the item.	The item is in front. Hands are free.	The agent has the item.
To put the item in the position.	The position is open. The agent has the item. In front of the position.	The item is at the position.

Table 3: *Examples of Order.*

Order	Target state
To take.	The agent has the item.
To move.	In front of the position.
To put.	The item is at the position.

To control the agent's action, the dialogue manager should know the agent's current position, the content of

action and the user's request on the action. To handle these, three groups, State, Action, and Order, are defined. Each entry of these three groups has its linguistic expression, which is necessary to generate sentences. "State" indicates the current state and has entries such as shown in Table 1. "Action" changes the initial state to the end state, and some of the entries are shown in Table 2. The agent changes its current state to the target state following to "Order." It has entries as shown in Table 3.

In order to decide the agent's action, an Order is first extracted from the user input. Then an Action is selected, whose end state coincides with the target state of the Order. If the selected Action's initial state is fulfilled, the agent completes the Action. If not, the initial state becomes the new target state, and the above procedure is repeated. By realizing all the target states, the agent completes the original Order from the user. As already mentioned, when any problems hard to be solved only by the agent arise, the agent tries to solve them through a dialogue with the user.

User's inputs are recognized by the system by referring to the dialogue data in the word dictionary. The dialogue data specifies the attributes of the words; such as word "red" is to represent the color of the item (item_color), word "take" is to represent the agent's action (agent_action), and so on. If the root phrase of the input sentence includes a word of "agent_action," the system recognizes the sentence being the user's order to the agent.

5. Sentence generation

The sentence generation can be done smoothly by representing linguistic information (concept) in LISP from with tags. To distinguish this from the word LISP form, we call it the tag LISP form. For instance, using the tags, the concept of putting an item in a position can be written as:

(oku \$PRED(o(\$ITEM))(ni(\$POS)))

Here, the tags \$ITEM, \$POS and \$PRED depict an item, a position and a predicate, respectively. The \$PRED tag means that the word "oku" works as a predicate when it is placed in a sentence.

Given a tag LISP form, a sentence (with syntactic structure) can be generated by pasting words and/or phrases at tag positions. Words conveying important information are decided by referring to the tags and the preceding user utterance.

The sentence generated by the dialogue manager should include information necessary for speech synthesis. For this purpose, the generated sentence is not in the orthographic form, but is a sequence of phone and prosodic symbols. The prosodic symbols are those we have formerly developed in the research works on dialogue speech [7]. Using "importance of word" and syntactic structure, we can automatically obtain the prosodic symbols for speech synthesis. The prosodic symbols decide the location of the phrase and accent commands for the generation process model, and the sentence F_0 contour can be generated. For instance, the symbol sequence for the sentence "terebino tokoroni idoudekinainodesuga dousureba iideshouka (I cannot move to the television set. What should I do?)" is as follows:

P121221 D411 te A0 re bi no to F411 ko ro ni A0 i F421
do u de ki na i A0 P21 D414 no A0 de su ga P22 D413 do
A0 u D412 su A0 re ba D413 i A0 i de sho u ka P0 S1

Here, the symbols starting with P show the phrase command locations and magnitudes. Accent command locations and amplitudes are shown by the symbols starting with D and F.

When generating a sentence for the dialogue to specify the item, the result of the item search in the room is converted to a phrase and connected to other phrases to construct a sentence. For instance, if there are several telephones and the system cannot specify, the generated sentence will be:

\$ITEM-wa ikutsuka aru \$SNTC-nodesuga \$SNTC-dorenokotodesuka

By cutting the phrase "deNwa (telephone)" from the user's input sentence (keeping its syntactic structure) and pasting it in the \$ITEM position, we can have the sentence "deNwawa ikutsuka arunodesuga dorenokotodesuka (There are several telephones. Which one are you talking about?)." A similar process is done for the dialogue to specify the position.

For the dialogue to decide the agent's action, sentences are generated from the linguistic information of State, Action, and Order. For instance, the Action "to put" has the linguistic information "To put \$ITEM to \$POS." Although it is possible to make all the thinking process of the agent into sentences, they are quite redundant. So the output sentences are only the final part of the sentences. For instance, if all the process is represented, the output sentences are:

I have ordered to put the black telephone in front of the television set. (Order)

But I do not have the black telephone. (State)

So, I thought to have the black telephone. (Action)
However, I am not in front of the black telephone. (State)

So, I thought to move to the front of the black telephone. (Action)

But, I cannot move to the front of the black telephone. (State)

So, what should I do?

However the actual output sentences from the system will simply be:

"I cannot move to the front of the black telephone. So, what should I do?"

6. Dialogue example

As the results, the following dialogue is conducted between user (U) and the agent (S):

U1: deNwao pasokoNnomaeni oite (Put the telephone in front of the personal computer.)

S1: deNwawa ikutsukaarunodesuga dorenokotodesuka (We have several telephones. Which one are you talking about?)

U2: kuroi deNwadesu (The black telephone.)

S2: deNwanotokoroni idoudekinainodesuga dousureba iideshouka (I cannot move to the telephone. So, what should I do?)

U3: kabiNo motte (Take the vase.)

S3: tega aiteinainodesuga dousureba iideshouka (My hands are not free. So, what should I do?)

U4: terebino maeni oite (Put it in front of the television set.)

S4: kabiNo oite iidesuka (Can I put down the vase?)

U5: hai (Yes.)

The underlined words are important words and prosodic emphases are placed there in speech output.

7. Conclusion

In an agent dialogue system newly developed, a method was realized to generate output speech directly from the reply content. Although the current system is a simple one and the output sentences have no varieties, the developed method of speech reply generation is a general one and can easily be applied for other spoken dialogue systems. Important words in a generated sentence were decided simply by referring to the tag information. A more detailed control should be developed. Also, there are cases where the prosodic control to emphasize important words is difficult, because of various restrictions on the prosodic features, such as requirements to represent syntactic structures. In such cases, word order of the sentence comes an important issue.

Ellipsis and anaphora are not studied in the current paper. We already have developed a simple method for their control by referring to the dialogue record [8]. We are planning to include the knowledge of the agent and the user for the control in the agent system.

This work was partly supported by Grant-in-Aid for Creative Basic Research (13NP0301).

8. References

- [1] Kiriya, S. and Hirose, K., "Development and evaluation of a spoken dialogue system for academic document retrieval with a focus on reply generation," *Systems and Computers in Japan*, Vol.33, No.4, pp.25-39 (2002).
- [2] Kiriya, S., Hirose, K. and Minematsu, N., "Prosodic focus control in reply speech generation for a spoken dialogue system of information retrieval," *Proc. IEEE Workshop on Speech Synthesis*, in CD-ROM (2002).
- [3] Shinyama, Y., Tokunaga, T., and Tanaka, H., "Kairai - Software Robots Understanding Natural Language," *Proc. 3rd International Workshop on Human-Computer Conversation*, pp.196-206 (2000)
- [4] Tago, J., Hirose, K. and Minematsu N., "Dialogue processing and output generation for an agent dialogue system," *Proc. 65th National Convention, Information Processing Society of Japan*, Vol.5, pp.507-510 (2003). (in Japanese)
- [5] Kawahara, T. et. al., "Product software of continuous speech recognition consortium -2001 version-," *SIG Notes, Information Processing Society of Japan*, 2002-SLP-43-3, pp.13-18 (2002). (in Japanese)
- [6] Fujisaki, H. and Hirose, K., "Analysis of voice fundamental frequency contours for declarative sentences of Japanese," *J. Acoust. Soc. Japan (E)*, Vol.5, No.4, pp.233-242 (1984).
- [7] Hirose, K., Sakata, M., and Kawanami, H., "Synthesizing dialogue speech of Japanese based on the quantitative analysis of prosodic features," *Proc. International Conference on Spoken Language Processing*, Vol.1, pp.378-381 (1996).
- [8] Hirose, K. and Senoo, T., "A method of generating speech reply with elliptical expressions and prosodic emphases," *Proc. ESCA Tutorial and Research Workshop on Spoken Dialogue Systems*, pp.233-236 (1995).