Nishimura *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2018) 2018:17
https://doi.org/10.1186/s13636-018-0142-8

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

Web-based environment for user generation of spoken dialog for virtual assistants

Ryota Nishimura[1*], Daisuke Yamamoto[2], Takahiro Uchiya[2] and Ichi Takumi[2]

Abstract

In this paper, a web-based spoken dialog generation environment which enables users to edit dialogs with a video virtual assistant is developed and to also select the 3D motions and tone of voice for the assistant. In our proposed system, "anyone" can "easily" post/edit contents of the dialog for the dialog system. The dialog type corresponding to the system is limited to the question-and-answer type dialog, in order to avoid editing conflicts caused by editing by multiple users. The spoken dialog sharing service and FST generator generates spoken dialog content for the MMDAgent spoken dialog system toolkit, which includes a speech recognizer, a dialog control unit, a speech synthesizer, and a virtual agent. For dialog content creation, question-and-answer dialogs posted by users and FST templates are used. The proposed system was operated for more than a year in a student lounge at the Nagoya Institute of Technology, where users added more than 500 dialogs during the experiment. Images were also registered to 65% of the postings. The most posted category is related to "animation, video games, manga."
The system was subjected to open examination by tourist information staff who had no prior experience with spoken dialog systems. Based on their impressions of tourist use of the dialog system, they shortened the length of some of the system's responses and added pauses to the longer responses to make them easier to understand.

Keywords: Spoken dialog system, MMDAgent, Digital signage, Real field, Web service, User-generated content, Consumer-generated media

1 Introduction

Spoken language processing technology has steadily improved over the years, and many commercial spoken dialog systems are now widely used by the public, such as Amazon's Alexa[1], Apple's Siri[2], Google's Google Assistant[3], and Microsoft's Cortana[4]. Engineers and researchers must create spoken dialog components for these virtual assistants, such as dialog scenarios, 3D characters, character motions, images, and character voices. Having engineers or researchers develop complex spoken dialog scenarios has the advantage of allowing people with professional knowledge and expertise to create them. On the other hand, this approach has the following drawbacks:

1. It may not always be possible for experts to create spoken dialog content from the viewpoint of the users. Engineers and researchers may not be able to successfully anticipate and satisfy users' requests.
2. It is not always possible for support staff to immediately respond to requests from users and system administrators when there are problems. It is difficult for engineers and researchers to manage these systems at all times and to deal with problems that can occur anywhere in the world.
3. As spoken dialog systems are more widely used, it may be difficult for a limited number of engineers and researchers to create all of the necessary content for these systems.

In order to solve these problems, in this study, we open the scenario creation process to general users or to the staff who manage spoken dialog systems at

*Correspondence: ryota@nishimura.name
[1]Department of Technology, Industrial and Social Science, Tokushima University, Tokushima, Japan
Full list of author information is available at the end of the article

actual installation sites. Since a larger number of people are involved in creating spoken dialog content, a wider variety of more appropriate content will be created. In other words, the idea of using user-generated content, as is being done with projects such as Wikipedia, Yelp, YouTube, and discussion blogs, was applied to spoken dialog scripts and other features of virtual assistants.

The purpose of this study is to apply the concept of user-generated content to spoken dialog systems and to construct an environment where ordinary people, such as general users or local administrators, can easily generate spoken dialog content. We deployed the proposed system experimentally to verify the effectiveness of such a content generation environment. Since user satisfaction is of key importance, we used this as our method of evaluating the viability of this approach.

The problems we encountered while developing the proposed system, and our proposed solutions, are described below:

- **Problem 1**: Any user must be able to create and edit the system's spoken dialog content. Furthermore, in order to create more attractive content, users should also be able to modify the assistant's tone of voice, 3D motion, and so on.
- **Problem 2**: If the same spoken dialog content file is edited by many users, it may be difficult to maintain consistency. There could also be technical problems such as double-editing (conflict of edited contents by simultaneous editing) and deadlock (a state in which two or more processing units wait for the completion of mutual processing and as a result any processing can not be advanced).
- **Problem 3**: In spoken dialog systems where users submit the content, many edits will likely be posted, but how will other users be notified of newly created content?
- **Solution to problem 1**: Provide a mechanism to edit spoken dialog and other content using web service technology.
- **Solution to problem 2**: Constrain the system's dialog to question-and-answer dialogs. Only allow the user who suggested the change or the local administrator to edit or revoke changes.
- **Solution to problem 3**: Display balloon panels for notifying other users of new content.

MMDAgent[5] [1] is a toolkit for building spoken dialog systems and was adopted as the base system of our spoken dialog system environment.

We carried out both practical and empirical research through experiments in which we allowed users to modify the proposed system. Based on the idea that a good spoken dialog system requires good content, our goal was to have users create high-quality spoken dialogs and other content.

2 Related work

Spoken dialog systems have previously been used in real-world environments, for example, Takemaru-kun [2] and Kita-chan [3], which are spoken dialog systems that can engage in simple Q and A dialogs. These systems have been in use for 10 years and have been used to collected data on natural human-machine interaction.

One method of managing spoken dialog systems is through the use of finite-state transducers (FSTs), a method which has been studied and developed by several researchers. Damnati et al. [4, 5] used an FST for speech recognition, spoken language understanding (SLU), and dialog control in a spoken dialog system for telephone applications. By integrating these FSTs, better results can be obtained than if the data is sequentially processed. ITSPOKE [6] is an Intelligent Tutoring SPOKEn dialog system, which is a speech-enabled version of the Why2-Atlas [7] text-based dialog tutoring system. In the Why2-Atlas system, the nodes of the FST are questions to the students, and the links exiting each node correspond to expected responses to the question. Hori et al. [8] used a weighted finite-state transducer (WFST) for SLU and dialog management. The WFST for SLU was composed of a word-to-concept WFST for language understanding, which was then optimized. Their study confirmed that the WFST-based dialog manager could accept recognition results from a speech recognizer well. They also have constructed a prototype spoken dialog system which functions as a Kyoto tour guide, using their WFST-based dialog system platform. The spoken dialog management system used in the present study is also based on a FST.

There have also been many studies and standardization proposals on how to describe dialog scenarios for spoken dialog systems. VoiceXML[6] [9, 10] is used for the development and distribution of speech applications and can be described as being the same as writing visual applications using HTML. A VoiceXML document is interpreted by the voice browser similar to the way an HTML document is interpreted by a Web browser. By using VoiceXML, it is possible for developers to define the various functions necessary for voice web services, such as the user's utterance grammar and dialog control. Typical applications of VoiceXML include voice guidance by telephone. Araki et al. [11] proposed a semi-automatic dialog system generator based on information found on the Internet, and they use a dialog generator to translate XML-based websites into VoiceXML. They also developed a frame-based spoken dialog system [12] using collected dialog scripts (VoiceXML).

One problem with using VoiceXML is that the available modalities are limited to voice and DTMF[7], so it is difficult to deal with multimodal data. One approach is to change the language specifications in order to expand the modality of the program. Multi-modal user interface description languages such as SALT [13] and XHTML [14] have been formulated; however, these languages are designed to add speech interaction to graphic web pages by adding spoken dialog descriptions to HTML codes and are not suitable for describing virtual agent interactions. Katsurada et al. [15] developed the multimodal dialog description language XISL[8], the modalities of which can be easily expanded, so that there is no need to change the language specifications. Since only the descriptions of interactions are defined separately from the XML content, reusability is improved. In addition, Katsurada et al. have developed Interaction Builder (IB)[9] [16, 17], which is a descriptive tool for dialog control that uses GUI. Each component that is selectable when using the GUI tool is described using XISL. IB employed a similar interface and functions as the RAD (rapid application development) tool provided in the CSLU[10] toolkit [18, 19], which is a well-known tool for constructing agent-based speech applications. In our research for this paper, we adopted GUI as our dialog description method, as in the research described above, but we restricted editable items to only basic items, since the users of the proposed application are ordinary people who are unfamiliar with dialog systems. The number of editable items available in previous studies would make the process too complex for ordinary people to navigate.

Learning-based dialog control construction methods such as POMDP[11] [20] have also been proposed. When using this method, input from the user is partially observed[12], and the state of the user is stochastically expressed by a pair of probabilities that indicate the state of the system. When POMDP is used for spoken dialog systems, training data are required. Although the efficiency of the training process has been improved, the basic design of the system is difficult.

Henderson et al. [21] used a neural network as a learning-based dialog control construction method. Recurrent neural networks (RNN) were used to map user input directly from the speech recognition results to the dialog state. Zhao et al. [22] developed an end-to-end framework for task-oriented dialog systems using Deep Recurrent Q-Networks (DRQN), such that the output utterances of the system were directly estimated from the input utterances of the user. A neural network-based model can be constructed easily if training data are available, but large amounts of data are necessary to achieve a high degree of accuracy.

We have proposed several methods of editing spoken dialog content. EFDE [23] is an interface that enables intuitive editing of state transition diagrams of spoken dialog content on tablet PCs, using touch panels and speech. One problem, however, is that if a state transition diagram is edited directly, the number of states increases and operability decreases. As a possible solution, we prepared several templates and treated each template as a state in order to reduce the number of states displayed. Although this makes it possible to edit more complex states than with the original system, it is difficult to allow simultaneous editing by multiple users. MMDAE [24] is a web service type interface that makes it possible to edit FST dialog scenarios directly on the web, making it possible to edit and share FSTs with a high degree of flexibility. However, this requires advanced knowledge about the FST. "Main entrance Mei-chan[13]" is a system that can register event information using a web browser. The system we are proposing in this paper is a developmental improvement of the "Main entrance Mei-chan" system.

3 MMDAgent toolkit

MMDAgent is a toolkit for building spoken dialog systems which integrates the functions necessary for constructing a spoken dialog system, such as automatic speech recognition (ASR), text-to-speech (TTS) synthesis, dialog control, 3D model rendering, and 3D model control, as shown in Fig. 1. It is possible to run MMDAgent on PCs [1] and smartphones [25] using operating systems such as Windows, Mac OS, Linux, iOS, and Android. MMDAgent uses Open JTalk [26] as its speech synthesis engine, Julius [27] as its speech recognition engine, MikuMikuDance [28] as its 3D model format, and Bullet Physics [29] as its physical operation engine. These functions are implemented as

Fig. 1 Example of a spoken dialog system created with MMDAgent

plug-ins, and they work together by exchanging messages with each other via the Global Message Queue.

MMDAgent is operated using spoken dialog scenario script files in FST format and related materials such as virtual agent 3D-CG, motion data and the image data to be displayed during dialogs, as well as acoustic and language models for the ASR and TTS. Materials and models related to the spoken dialog scripts are called spoken dialog content. When using MMDAgent, multiple FST scripts can operate in parallel and independently, but when multiple FST scripts are operating in parallel, there is the possibility of deadlock, etc., due to interference between resources and messages, so the FST scripts need to be carefully designed.

3.1 FST scenario scripts

An example of an FST script is shown in Fig. 2. In each script, a state transition machine is described and events are generated from the functional components of the system (speech recognition, speech synthesis, 3D model, variables, dialog manager, and plugins) as inputs and the commands to the functional components are the outputs. The events detected by the system (FST inputs) and the commands executed by the system (FST outputs) are as follows:

- "Add, change, and delete" information related to the 3D model
- "Add, change, delete, and change speed" of the motion of the 3D model
- "Change coordinates and angle" of the 3D model
- "Play or stop" the sound
- "Load and display" the stage (floor and background)
- "Set color and direction" of the light source
- Camera movement
- "Set, delete, and compare" variables
- "Start or stop" the timer
- "Enable or disable" plug-ins
- "Start and end" of speech recognition events
- "Play or stop" speech synthesis events
- "Change" ASR gain, dictionary, and config file in commands
- "Play or stop" speech synthesis, or lip sync in commands

In Fig. 2, when the speech recognition function unit recognizes the word "Hello," the system transits from state 1 to state 10. Next, the "motion start" command (MOTION_ADD) and the "speech synthesis start" command (SYNTH_START) are output and transition to state 12 occurs. Until the playing of the synthesized speech is finished, the system remains in standby state (state 12), and when the "speech synthesis end" event (SYNTH_EVENT_STOP) occurs, the system transitions to state 1.

Because FSTs are considered to be the equivalent of an automaton with outputs, complicated controls such as interruption (e.g., barge-in), processing according to context, and sequential dialog control can be realized using an FST. On the other hand, it is difficult to describe these complex scripts manually on a large scale; thus, it is necessary to develop techniques such as automated script generation using the database and dedicatedk tools.

4 Proposed system

As shown in Fig. 3, the proposed system consists of a spoken dialog sharing service (a web server) and an FST generator. The proposed system was developed using Java, PostgreSQL was adopted as the database, and Apache Click was adopted as the web framework.

Anticipated problems were the following:

1) Detailed information can be difficult to convey using speech alone, and

```
 1  10  RECOG_EVENT_STOP|Hello            <eps>
 1  10  RECOG_EVENT_STOP|Good morning     <eps>
10  11  <eps>                            MOTION_ADD|mei|greet|greet.vmd
11  12  <eps>                            SYNTH_START|mei|normal|Hello
12   1  SYNTH_EVENT_STOP|mei              <eps>
```

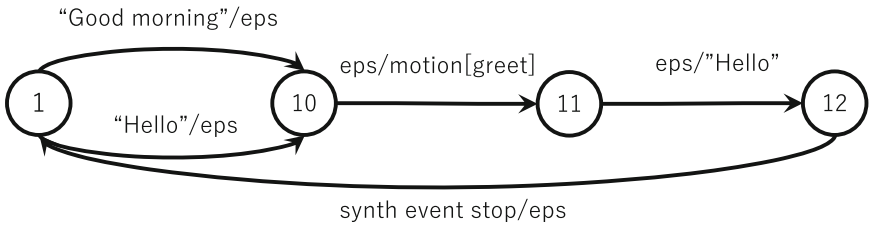

Fig. 2 FST Script (top) and its state transition diagram (bottom)

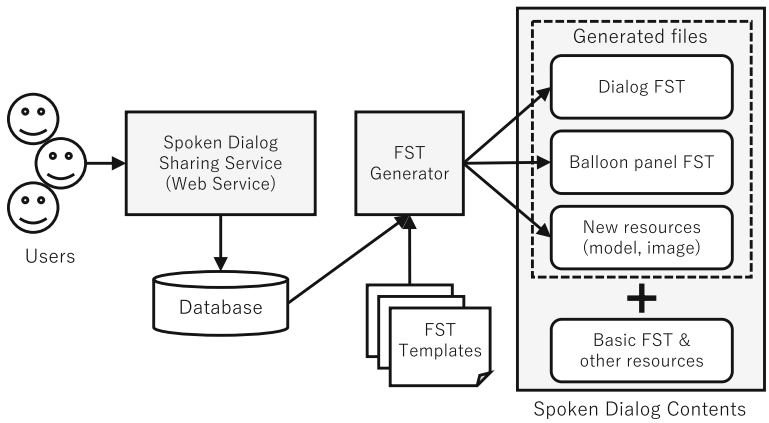

Fig. 3 System configuration

2) Users may be unaware of what topics they can talk about.

To solve problem 1, when the system explains something using speech, related images are displayed to help users understand. To solve problem 2, we adopted a balloon panel system, which is described in Section 4.7.

4.1 Spoken dialog model

The spoken dialog model used by the proposed system is a set of question-and-answer type dialogs. When speech which includes one of several keywords is input, a response corresponding to the keyword is generated and relayed to the user. This response operation includes several commands, as described in Section 3.1.

4.2 Spoken dialog sharing service

The spoken dialog sharing service allows many users to edit or contribute to the contents of the interaction using a standard web browser installed in a smartphone or a PC. In the proposed system, the spoken dialog contents which can be edited are restricted to typical question-and-answer type dialogs. As a result, even if multiple users edit the FST independently and in parallel, double editing and deadlock problems can be avoided.

Users can post content of the following types, as shown in Fig. 4:

- Input sentence keywords
- Pronunciation of input
- Output response sentence (two kinds)
- Speech synthesis settings
- Facial expressions of the virtual assistant
- Motion of the virtual assistant
- Image for display in balloon panels

The posted information is then stored in the database.

Users need to log in to the editing service on the web. It is also possible to log in using an external account such as a Google account. Figure 4 shows the interface for posting a dialog pattern on the web. The dialog shown in Fig. 5 is entered as the target dialog.

"Keywords" are those used to elicit question-and-answer type dialogs, and "pronunciation" describes how the keyword is pronounced using hiragana (phonogramic Japanese characters). The screen shown in Fig. 4 is used to add the topic keyword and its pronunciation to the language model for the ASR system. In the "response" field, input text for the text-to-speech synthesizer is described. This is the user-generated response the virtual assistant should give when inquiries include the keyword. For "voice," the virtual assistant's tone of voice can be selected, and "facial expression" and "motion" can be used to select the movements of the 3D characters. An image to be displayed on the balloon panel can be uploaded to the "keyword panel image," and an image to be displayed together with a speech response is input to the "poster panel image."

There are two input slots for system responses. Response 1 is spoken by the virtual assistant with standard voice and motion. Response 2 is synthesized with the tone of voice and motion selected by the user. We thought that generating two kinds of response sentences, with different tones of voice and body movements, might improve the charm of the spoken dialog. When a dialog pattern is posted, it is stored in the database and added to the dialog pattern list. Each dialog pattern can be modified, deleted, etc. only by the user who posted it and by the local administrator.

4.3 FST templates

The FST templates are for generating FST scripts based on the dialog content posted by users. Templates for dialog (with both image and non-image versions) and templates

Fig. 4 Web screen allowing users to edit the system

USER: 名古屋市科学館に行きたいな.
(I would like to go to Nagoya City Science Museum.)
SYS: 名古屋市科学館のプラネタリウムは世界一の大きさです.
(The planetarium of the Nagoya City Science Museum is the largest in the world.)
すごいですよね.
(That's amazing. : **with Voice, Motion, Facial expression**)

Fig. 5 Example of a dialog provided by a user

```
 1   10   RECOG_EVENT_STOP|Keyword      <eps>
10   11   <eps>                         SYNTH_START|mei|normal|Response 1
11   12   SYNTH_STOP|mei                SYNTH_START|mei|Voice|Response 2
12   13   <eps>                         MOTION_ADD|mei|action|Motion
13   14   <eps>                         MOTION_ADD|mei|expression|Face
14    1   SYNTH_STOP|mei                <eps>
```

Fig. 6 Example of FST template for dialog FST (non-image version)

```
 1   10   RECOG_EVENT_STOP|名古屋市科学館          <eps>
10   11   <eps>                                  SYNTH_START|mei|normal|名古屋市科学館のプラ...
11   12   SYNTH_STOP|mei                         SYNTH_START|mei|happy|すごいですよね
12   13   <eps>                                  MOTION_ADD|mei|action|surprise.vmd
13   14   <eps>                                  MOTION_ADD|mei|expression|normal.vmd
14    1   SYNTH_STOP|mei                         <eps>
```

Fig. 7 FST generated from input in Section 4.2

for displaying balloon panels are prepared. An example of an FST template (not using images) is shown in Fig. 6. Bold and underlined words (keyword, response 1, response 2, motion, voice, and facial expression) are components that can be changed by the FST generator. In the template for the display of balloon panels, the balloon panel is displayed randomly and the image file name input by the user is added to the template. The interaction template (using images) includes a command for displaying the target balloon panel at large size.

4.4 FST generator

The FST generation function automatically generates FST scripts from information stored in the database collected by the spoken dialog sharing service. Interaction content available to MMDAgent is created by combining the database and the FST template. The 3D model, motion for the 3D model (body movement and facial expressions), and the acoustic model for each emotion are contained in the "basic resources" of MMDAgent (see Section 4.6), so even if these models are used in the template, there is no need for the user to upload them back to the database again.

As an example of automatic FST generation, if the system uses the data input by the user as shown in Fig. 4 and an FST template, the FST shown in Fig. 7 will be generated. The automatically generated FST is based on information on spoken dialog content stored in the database. We adopted the FST template method [23] as our automatic generation method, which insures that the FST templates are used properly according to the type of dialog being created, and values from the database are applied to the variables within the template. However, since it is necessary to maintain consistency among the state numbers when using an FST, the state number is managed by an automatic generation function unit. There is also a mechanism to automatically generate two FST scripts, one for speech recognition and one for the balloon panel. The system also supports barge-in by controlling the format of the FST template.

4.5 Generated files

The three kinds of data described in this section (dialog FST, balloon panel FST, and new resources) are output by the FST generator using the dialog patterns in the database collected by the spoken dialog sharing service and the FST template. Regarding "news resources" (images and 3D/motion/ASR/TTS models submitted by users), the FST generator does not change the images or models uploaded by the user, nor does it change or generate other images or models.

Rules for matching user input with keywords in the speech recognition result and for determining the corresponding system output (speech synthesis output) are defined in the **dialog FST**.

In the **balloon panel FST**, FST scripts for displaying the images posted by users are described, and the images are randomly displayed on the screen like fluffy bubbles.

Images and models submitted by users are output as "new resources," and a dictionary file for new words is generated using the words and pronunciation from the user input. This user-provided pronunciation is used as

Fig. 8 Presentation of interactive content in icon format

Fig. 9 Balloon panel in poster format

the word pronunciation information when the keyword is not included in the language model for speech recognition. Specifically, the keyword and user-provided pronunciation are added to the Julius ".dic file" for new word registration.

4.6 Basic FST and other resources

In addition to the files generated by the FST generator, statically created FST scripts (basic FST) can also be used in our spoken dialog system. Basic FSTs include greetings, self-introductions, weather forecasts, and fortune-telling as basic dialogs. These basic FSTs are generally intended to be manually defined by an engineer. As mentioned above, our automatic generation function is limited to generating only question-and-answer type dialogs, but since many types of dialogs can be created using the basic FSTs, complicated dialogs can also be created.

"Other resources" include the images, 3D models, acoustic, and language models for speech recognition and the speech synthesis model used in the basic FST. The following models are prepared for the dialog agent for "Mei-chan" and "SD Mei-chan"[14] (a miniaturized version of "Mei-chan"):

- Five kinds of facial expressions (anger, bashfulness, happiness, listening, and sadness)
- Five kinds of voices (angry, bashful, happy, normal, and sad)
- 25 kinds of body motions

4.7 Balloon panel

One problem some users have when using spoken dialog systems is that they do not know what to say to the system. As a solution to this problem, the proposed system displays images showing keywords which are recognized by the system. In Figs. 8, 9, and 10, icon type, poster type, and keyword (cloud form) type panes are displayed, respectively. The icon type and the poster type panels are already supported by the standard version of "Mei-chan," but users need to come up with speech recognition keywords based on these image panels, which is difficult for ordinary, novice users. So in this study, we propose an alternative to "image only" keyword indication, which is allowing users to read keywords displayed on the panel, making it easier for them to talk to the system. We also make it easier for users to imagine the content of the dialog by displaying images together with keywords. Because it is difficult to display many keywords at the same time, as shown in the figures, the keywords are displayed one by one in alternating balloon panels. One effect of using the balloon panels is that the system becomes easier to use for children, the elderly, and handicapped persons with poor

Fig. 10 Balloon panel in keyword format

Fig. 11 Proposed system at Handa Tourist Information Center

speech recognition rates, since they can interact with the system by touching the balloons using the touch panel.

5 Experiment and evaluation

To test our proposed system, we developed and operated several demonstration systems and experimented with our user-generated spoken dialog environment. The locations where the development system was used are listed in Section 5.1. An analysis of the results of the experimental installations at the "tourist information center" and "university student lounge" is shown in Sections 5.2 and 5.3.

5.1 Demonstration experiment sites

5.1.1 Tourist information center

Our spoken dialog system was set up at the Handa City Tourist Information Center in Aichi Prefecture. The system was used to provide tourists with information about Handa City, and the staff of the tourist information office provided content for the system.

5.1.2 University student lounge

The system was also set up in a student lounge at the Nagoya Institute of Technology. All of the students were able to post dialog content and images by logging in with an account provided by the university.

5.1.3 Public event at a TV station

The system was used by the public at an event run by the Nagoya Bureau of the NHK television network, and the TV station's staff provided content and managed the system.

5.1.4 Information station at a conference

The proposed system was used for session guidance at the Tokai Joint Conference on Electrical, Electronics, Information, and Related Engineering in 2017. In addition to dialog keywords, the balloon panels also displayed information such as session numbers, dates, and times. In addition to providing information using speech synthesis, a map showing the route to the room where their session was being held was also displayed.

5.1.5 City hall

Our spoken dialog system, with a larger display, was installed at Handa City Hall in Aichi Prefecture.

5.2 Installation at the Handa City Tourist Information Center

In February 2014, we set up our spoken dialog system at the Handa City Tourist Information Center, as shown in Fig. 11. The system was used to provide tourist

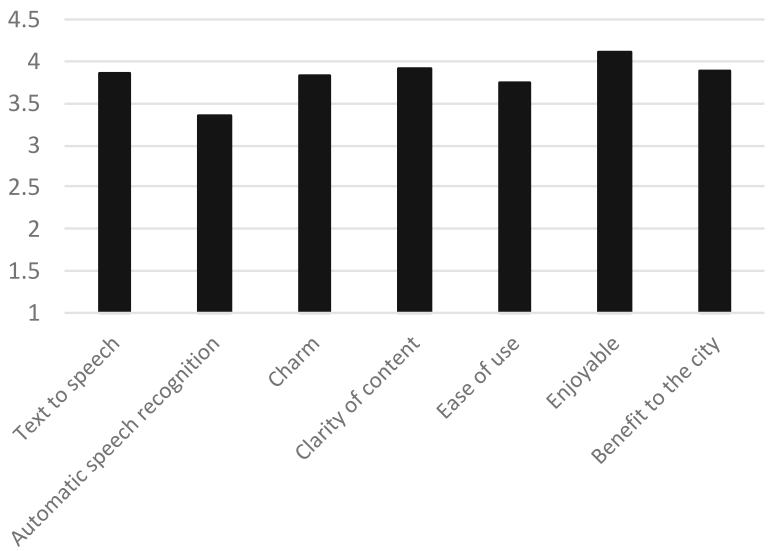

Fig. 12 Questionnaire results from Handa tourists "clarity of content" "benefit to the city"

Fig. 13 Proposed system at Nagoya Institute of Technology

information about Handa City, and the staff of the tourist information office created the content for the dialog system.

For the first version of the system, photographs and descriptions of tourist spots suggested by the staff of the tourist information office were input, and we installed 22 dialogs. The tourism staff then deleted 6 of our dialogs, updated the contents of 15 other dialogs, and added 6 new dialogs. The sentences of the original dialogs contained more characters since the original dialog sentences were longer, and the standard deviation for original sentence length was also larger (91.8 characters per sentence on average, with a standard deviation of 29.3 characters). After the updates by the tourism staff, the lengths of the sentences and their standard deviation were both smaller (73.1 characters per sentence on average, with a standard deviation of 16.3 characters). In addition, the following adjustments were made by the staff:

- Commas were added between the words in the virtual assistant's responses to improve the ease of understanding the system's synthesized speech.
- The written responses were changed to spoken responses.
- Changes were made to the character's motion and voice.

These results show that by introducing a mechanism for users (in this case, the staff of the tourist information office) to update the system on-site, better spoken dialog systems can be realized.

In addition, a questionnaire was given to tourist information office visitors who used the system. They were asked to rate various aspects of the system on a scale from 1 to 5, and usable responses were obtained from 39 people. Questionnaire results are shown in Fig. 12. Overall, their responses were positive, and when asked if the system was "enjoyable" to use, the system received an average rating of 4.1. There were also many positive comments, such as, "There were many variations in the replies" and "It was good and enjoyable to use." The system function which allowed local administrators to change the character's voice and body movement can also be considered to have been effective. On the other hand, there were also suggestions for improvement:

- It was difficult to understand responses given only with speech.
- It would be better to have information such as a map or a picture when explaining.
- It would be better if we could respond using a touch panel. (This version of our display did not include a touch panel.)

Pictures were displayed on the balloon panel, but some users felt the information provided was still inadequate.

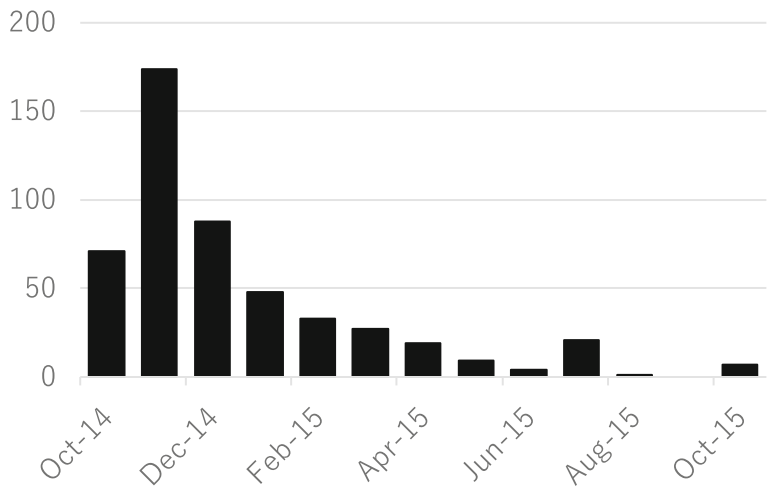

Fig. 14 Number of submissions per month

5.3 Installation in the university student lounge

Our system was also installed in a student lounge at the Nagoya Institute of Technology, as shown in Fig. 13. All of the students were able to post dialog by logging in with an account provided by the university. Formal operation began on October 1, 2014, and 502 posts were made by October 22, 2015. Figure 14 shows the number of postings per month. An additional 28 changes were posted by local system managers during initial operation, so there were a total of 530 dialog changes in all. The total number of unique users was 47. As shown in Table 1, one user posted 363 times, while half of the users (24 people) posted only once.

Posting images for balloons was optional, but 65% of the posts (327 posts) included balloon images. Uploading of the balloon images was expected to be a burden to the users, but it seems to have been a natural act for many of them. On the other hand, as shown in Table 2, 78% of the voice settings and 80% of the motion settings remained "normal," which are the initial settings. Voice and motion settings may have remained set to normal because it was not possible for the users to preview their changes during editing. We hope to add this function to future versions of the system.

The average number of characters in the user-provided, synthesized dialog speech was 71.2 characters per sentence, with a standard deviation of 53.8. This is almost the same as the average number of characters for the improved responses at the Handa City Tourist Association (73.1 characters), but the standard deviation at the student lounge was extremely large. This is probably because the system in Handa City was restricted to tourist information, while the system installed at the university had a much higher degree of freedom with regard to topics. Some of the user suggested responses were longer than 300 characters per sentence. When the duration of the synthesized speech is long, user waiting time also becomes long, so the "barge-in" function was apparently effective. To deal with long, user-provided responses, the following countermeasures may be effective, but introducing them to the system is a future task:

Table 1 Number of persons who made various numbers of submissions

No. of sub.	Num. of persons
1	24
2	9
3	2
4	4
From 5 to 10	5
From 11 to 20	2
363	1

Table 2 Number of selections of various tones of voice and motions for characters by university students

Tone of voice		Motion type	
Normal	391	Normal	402
Happiness	63	Happiness	31
Anger	20	Cheer up	27
Embarrassment	16	Embarrassment	18
Sadness	12	Sadness	12
		Surprise	6
		Bye	3
		Greeting	3

- Summarizing the contents of the response
- Splitting the response into multiple dialogs
- Adjusting the speaking speed so the character talks quickly

The type of content posted is shown in Table 3. The number of posts related to "animation, video games, manga" was 151, which was the largest number by category.

We also investigated usage of user generated dialog by topic. Figure 15 shows the number of times each posted dialog was used. We can see a usage pattern of the "long tail" type, which is one of the features of user-generated content. The top-ranking content is shown in Table 4. Each posted dialog was used 11.4 times on average.

6 Conclusions

In this paper, we proposed a spoken dialog content-generation system which relies on user-generated content. Using web service technology, our system allows anyone to easily edit spoken dialog content and other virtual video assistant features such as tone of voice, facial expression, and body motion. By limiting the spoken dialog to question-and-answer type interactions, problems such as double editing and deadlock were avoided. We also

Table 3 Content generated by university students

Type of content	No. of submissions
Animation, game, manga	151
General knowledge	83
Chat	81
School	53
Music, entertainment	40
People	38
Shops, facilities	23
Science	8
Other, unknown	25

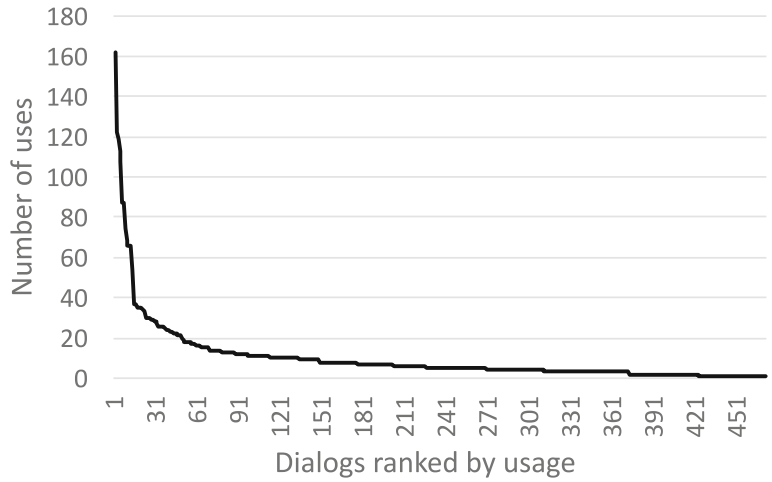

Fig. 15 Usage of spoken dialog provided by users

introduced a balloon panel system as a method of presenting additional, user-generated visual information. We reported detailed results of our demonstration experiments at the Handa City Tourism Association and the Nagoya Institute of Technology student space, and briefly described similar trials at a public event at the NHK TV station in Nagoya, at an academic conference and at Handa City Hall. At the installation at the Nagoya Institute of Technology, users posted over 500 dialogs; however, there was a typical long tail trend regarding usage of this user supplied content. In general, the proposed system was well accepted by users, who actively provided dialog and visual content to improve the performance of the system.

In this research, a method to effectively collect user-generated contents was proposed, and also surveys were conducted on the number of contributors and the content type. Investigation of the change in the performance of the spoken dialog system due to the change in the amount of posted contents is a future work.

Table 4 Highest ranking content by keywords

Keyword	Num. of use
Haruna (video game character)	162
Repeat a school year	122
Google (verb)	119
Louise (novel character)	113
Typhoon	108
Do not press	74
Hot air	68
Garchomp (video game character)	66
Tenhou (mahjong term)	66

As future work, we would like to investigate if recognition of the virtual assistant's speech by users is improved when the system simultaneously provides corresponding visual information. We will also explore a method of providing a poster image with the system's verbal response, as well as a method of allowing users to respond to the system using a touch panel.

Endnotes

[1] https://www.alexa.com/

[2] https://www.apple.com/ios/siri/

[3] https://assistant.google.com/

[4] https://www.microsoft.com/cortana/

[5] http://www.mmdagent.jp/

[6] Voice eXtensible Markup Language

[7] Dual-Tone Multi-Frequency

[8] eXtensible Interaction Sheet Language

[9] Interaction Builder is included in the Galatea toolkit for spoken dialog systems with an anthropomorphic agent

[10] Center for Spoken Language Understanding, Oregon Graduate Institute

[11] Partially Observable Markov Decision Process

[12] The user's intention and contents of their utterance are the targets to be observed, but accurate observation is impossible due to speech recognition errors. The system performs estimation for these states.

[13] http://www.nitech.ac.jp/eng/mei/

[14] Super Deformed

Acknowledgements
This study was supported by the Core Research for Evolutional Science and Technology (CREST) of the Japan Science and Technology Agency (JST), and by the Strategic Information and Communications R&D Promotion Program (SCOPE) of Ministry of Internal Affairs and Communications (MIC) of Japan.

Authors' contributions
RN and DY wrote the paper. RN, DY, and TU performed the experiments, and IT organized and revised the experiment. All of the authors discussed the final results. All of the authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1]Department of Technology, Industrial and Social Science, Tokushima University, Tokushima, Japan. [2]Department of Computer Science, Nagoya Institute of Technology, Nagoya, Japan.

Received: 1 March 2018 Accepted: 25 October 2018
Published online: 16 November 2018

References

1. A. Lee, K. Oura, K. Tokuda, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. MMDAgent - A fully open-source toolkit for voice interaction systems, (2013), pp. 8382–8385. https://doi.org/10.1109/ICASSP.2013.6639300
2. R. Nisimura, A. Lee, M. Yamada, K. Shikano, in *Proceedings of the 9th European Conference on Speech Communication and Technology (INTERSPEECH-2005)*. Operating a public spoken guidance system in real environment (ISCA, 2005), pp. 845– 848. http://www.isca-speech.org/archive/interspeech_2005. https://www.isca-speech.org/archive/interspeech_2005/i05_0845.html
3. H. Kawanami, S. Takeuchi, R. Torres, H. Saruwatari, K. Shikano, in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2011 (APSIPA2011)*. Development and operation of speech-oriented information guidance systems, kita-chan and kita-robo (APSIPA, 2011), pp. 558–561. http://www.apsipa.org/proceedings_2011/. http://www.apsipa.org/proceedings_2011/pdf/APSIPA243.pdf
4. G. Damnati, F. Béchet, R. De Mori, in *Proceedings of 2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*. Spoken language understanding strategies on the France telecom 3000 voice agency corpus (IEEE, 2007). https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4216989. https://doi.org/10.1109/ICASSP.2007.367150
5. C. Raymond, F. Béchet, R. De Mori, G. Damnati, On the use of finite state transducers for semantic interpretation. Speech Comm. **48**(3-4), 288–304 (2006). https://doi.org/10.1016/j.specom.2005.06.012
6. D.J. Litman, S. Silliman, in *Proceedings of the Human Language Technology Conference: 4th Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*. ITSPOKE: an intelligent tutoring spoken dialogue system, (2004), pp. 233–236
7. K. VanLehn, P.W. Jordan, C.P. Rosé, D. Bhembe, M. Böttner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, et al., in *International Conference on Intelligent Tutoring Systems*. The architecture of why2-atlas: A coach for qualitative physics essay writing (Springer, Berlin, 2002), pp. 158–167. https://doi.org/10.1007/3-540-47987-2_20
8. C. Hori, K. Ohtake, T. Misu, H. Kashioka, S. Nakamura, in *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH-2008)*. Dialog management using weighted finite-state transducers (ISCA, 2008), pp. 211–214. https://www.isca-speech.org/archive/interspeech_2008/. https://www.isca-speech.org/archive/interspeech_2008/i08_0211.html
9. VoiceXML Forum, VoiceXML: The standard application language for voice dialogues. http://www.voicexml.org/
10. W3C Recommendation 16 March 2004: Voice extensible markup language (VoiceXML) Version 2.0. https://www.w3.org/TR/voicexml20/
11. M. Araki, T. Ono, K. Ueda, T. Nishimoto, Y. Niimi, Ù. Óòó, in *In EUROSPEECH-2001*. An automatic dialogue system generator from the Internet Information Contents Department of Electronics and Information Science, (2001), pp. 1743–1746
12. M. Araki, in *Proceedings of the International Workshop on Semantic Web Foundations and Application Technologies (SWFAT)*. Owl-based frame descriptions for spoken dialog systems (SWAFT, 2003), pp. 1–2. http://www-kasm.nii.ac.jp/SWFAT/. http://www-kasm.nii.ac.jp/SWFAT/PAPERS/SWFAT04S.PDF
13. SALT Forum, SALT: Speech Application Language Tags. http://www.saltforum.org/
14. W3C Note 21 December 2001: XHTML+Voice Profile 1.0. https://www.w3.org/TR/xhtml+voice/
15. K. Katsurada, Y. Nakamura, H. Yamada, T. Nitta, in *Proceedings of the 5th International Conference on Multimodal Interfaces - ICMI '03*. XISL: a language for describing multimodal interaction scenarios (ACM Press, New York, 2003), pp. 281–284. https://doi.org/10.1145/958432.958483
16. K. Katsurada, H. Adachi, K. Sato, H. Yamada, T. Nitta, Interaction builder: A rapid prototyping tool for developing web-based MMI applications. IEICE Trans. Inf. Syst. **E88-D**(11), 2461–2467 (2005). https://doi.org/10.1093/ietisy/e88-d.11.2461
17. K. Katsurada, A. Lee, T. Kawahara, T. Yotsukura, S. Morishima, T. Nishimoto, Y. Yamashita, T. Nitta, in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2009 (APSIPA2009)*. Development of a Toolkit for Spoken Dialog Systems with an Anthropomorphic Agent: Galatea (APSIPA, 2009), pp. 148–153. http://www.apsipa.org/proceedings_2009/. http://www.apsipa.org/proceedings_2009/pdf/MP-SS1-5.pdf
18. F. Michael, in *Proceedings of the Sixth European Conference on Speech Communication and Technology (EUROSPEECH'99)*. McTear Software to support research and development of spoken dialogue systems (ISCA, 1999), pp. 339–342. https://www.isca-speech.org/archive/eurospeech_1999/. https://www.isca-speech.org/archive/eurospeech_1999/e99_0339.html
19. S. Sutton, R. Cole, J. De Villiers, J. Schalkwyk, P. Vermeulen, M. Macon, Y. Yan, E. Kaiser, B. Rundle, K. Shobaki, P. Hosom, A. Kain, J. Johan Wouters, D. Massaro, M. Cohen, in *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98)*. Universal Speech Tools: The CSLU Toolkit (Australian Speech Science and Technology Association, Incorporated (ASSTA), 1998), pp. 3221–3224. http://andosl.anu.edu.au/icslp98/icslp98_contents.html
20. J.D. Williams, S. Young, Partially observable Markov decision processes for spoken dialog systems. Comput. Speech Lang. **21**(2), 393–422 (2007). https://doi.org/10.1016/j.csl.2006.06.008
21. M. Henderson, B. Thomson, S. Young. Word-based dialog state tracking with recurrent neural networks, (2014), pp. 292–299
22. T. Zhao, M. Eskenazi, in *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2016)*. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning (Association for Computational Linguistics, Los Angeles, 2016), pp. 1–10. http://aclweb.org/anthology/W16-3601. https://doi.org/10.18653/v1/W16-3601
23. K. Wakabayashi, D. Yamamoto, N. Takahashi, in *A Voice Dialog Editor Based on Finite State Transducer Using Composite State for Tablet Devices*, ed. by R. Lee (Springer, Cham, 2016), pp. 125–139. https://doi.org/10.1007/978-3-319-23467-0_9
24. R. Nishimura, D. Yamamoto, T. Uchiya, I. Takumi, in *Proceedings of the Second International Conference on Human-agent Interaction. HAI '14*. Development of a dialogue scenario editor on a web browser for a spoken dialogue system (ACM, New York, 2014), pp. 129–132. https://doi.org/10.1145/2658861.2658904
25. D. Yamamoto, K. Oura, R. Nishimura, T. Uchiya, A. Lee, I. Takumi, K. Tokuda, in *Proceedings of the Second International Conference on Human-agent Interaction. HAI '14*. Voice interaction system with 3d-cg virtual agent for stand-alone smartphones (ACM, New York, 2014), pp. 323–330. https://doi.org/10.1145/2658861.2658874
26. HTS Working Group, Open JTalk: The Japanese TTS System. http://open-jtalk.sourceforge.net/
27. A. Lee, T. Kawahara, in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2009 (APSIPA2009)*. Recent development of open-source speech recognition engine julius (APSIPA, 2009), pp. 131–137. http://www.apsipa.org/proceedings_2009/. http://www.apsipa.org/proceedings_2009/pdf/MP-SS1-3.pdf
28. Y. Higuchi, MikuMikuDance: Vocaloid Promotion Video Project. https://sites.google.com/view/vpvp/
29. E. Coumans, Bullet physics library. http://www.bulletphysics.org/