Example Phrase Adaptation Method for Customized, Example-Based Dialog System Using User Data and Distributed Word Representations

Norihide KITAOKA† a, Member, Eichi SETO† †, Nonmember, and Ryota NISHIMURA† †, Member

SUMMARY We have developed an adaptation method which allows the customization of example-based dialog systems for individual users by applying “plus” and “minus” operations to the distributed representations obtained using the word2vec method. After retrieving user-related profile information from the Web, named entity extraction is applied to the retrieval results. Words with a high term frequency-inverse document frequency (TF-IDF) score are then adopted as user related words. Next, we calculate the similarity between the distributed representations of selected user-related words and nouns in the existing example phrases, using word2vec embedding. We then generate phrases adapted to the user by substituting user-related words for highly similar words in the original example phrases. Word2vec also has a special property which allows the arithmetic operations “plus” and “minus” to be applied to distributed word representations. By applying these operations to words used in the original phrases, we are able to determine which user-related words can be used to replace the original words. The user-related words are then substituted to create customized example phrases. We evaluated the naturalness of the generated phrases and found that the system could generate natural phrases.

key words: example-based spoken dialog system, user customization, distributed word representation

1. Introduction

The performance of natural language dialog systems has been improving over time, and applications using these systems are becoming more popular [2]. Various types of dialog systems have been developed, such as counseling dialog systems [3] and conversational knowledge teaching agents [4]. More recently, chat-like dialog systems are being actively investigated, since these systems will be needed for symbiotic machines such as humanoid robots. Reminiscence therapy systems also required this type of interactive dialog technology [1]. However, when designing a chat-like spoken dialog system, there are various problems which need to be resolved in order for the system to generate natural conversations. This study addresses response sentence generation, which is the generation of a natural response to a user’s utterance, which is one of the key problems in developing interactive spoken dialog systems. We develop an example-based, spoken dialog/chat system which is adapted to a specific user. Since systems such as reminiscence therapy dialog systems are used by a specified user, user adaptation can be used to improve performance. Example-based spoken dialog systems can robustly respond to user utterances if the example database contains a wide enough variety of utterances. However, it is more difficult to generate personalized responses which are related to a specific user’s interests or preferences, so our goal is to develop a database which is customized for each user. In [20], a domain adaptation for a sequence-to-sequence-based dialog system was investigated. Enrichment of sequence-to-sequence-based dialog generation using external memory has also been proposed [21]. In this study, we develop a method of creating an example database using information about a user acquired through a web search. When generating example phrases using the database, we propose using word2vec operations [5]–[9] to substitute user specific information into the example phrases [23]. Simply replacing words in example phrases with user-related words can result in inappropriate phrases, however. For example, “You’re from Tokyo? Have you visited the Skytree?” could become “You’re from Osaka? Have you visited the Skytree?.” Arithmetic operations over word2vec distributed word representations can resolve these kind of inconsistencies.

This paper is organized as follows In Sect. 2, we describe a typical example-based (spoken) dialog system. In Sect. 3, we explain how we obtained user-adapted dialog examples. We discuss the details of our proposed application of arithmetic operations to distributed representations of words during example generation in Sect. 4. We describe our experimental evaluation of our proposed method in Sect. 5 and conclude this paper in Sect. 6.

2. Example-Based Spoken Dialog Systems

An example-based dialog system is a dialog system that responds to a user by using an example database consisting of pairs of input examples and their corresponding responses. Figure 1 shows examples of such input and response pairs. If the inputs are speech rather than text, and the responses are also speech, the system is called an example-based spoken dialog system, and a speech recognizer and a speech synthesizer are used. Figure 2 shows the flow of an example-based spoken dialog system. Example responses may correspond to multiple example inputs, since various user ut-
utterances can have the same meaning, (e.g., “Do you know what time it is?”), “Do you have the time?”, “What time is it?”). By preparing multiple input examples, the system can respond to a wider variety of user utterances. Likewise, multiple example responses can be used in response to the same example input, (e.g., if asked, “How are you today?” the system can respond in different ways, such as, “Fine, thanks”, or “So-so”, or “I’m excited!”). By preparing multiple example responses the system can provide conversations which are less monotonous.

Typical spoken dialog systems recognize a user’s utterances using speech recognition software, and then match the content of the recognized sentence with the system’s input examples. Each input example is scored according to the number of matched content words, and after all of the input examples are scored the one with the highest score is selected. A score for each of the possible response sentences is then calculated. When a response sentence corresponds to multiple input sentences, the highest score among the corresponding input sentences is used for the score of the response sentence.

One successful example-based spoken dialog system is “Takemaru-kun” [10]–[12], which has been used with the public for long time. The system contains prepared responses such as greetings, self-introductions and directions for navigating community centers and the surrounding facilities. As various users talk to the system, the system logs these real-world dialogs, which are then transcribed and adopted as new examples, making the system more robust to an increasing variety of user utterances. Our proposed system does not adopt previous dialogs as examples, however, because its responses are customized for a specific user, and we do not anticipate an extremely wide range of possible utterances from one person. Instead, we customize the example database using a limited amount of information about the user.

There have been many studies on example-based spoken dialog systems. For example, a method to improve the robustness of an example-based dialog modeling framework using an agenda-based approach and n-best recognition hypotheses has been proposed [13]. A new probabilistic framework for spoken dialog management using a frame-based belief state representation has also been proposed [14]. These dialog management studies focus on selecting the correct response according to a given situation. In this study, we propose a method to adapt the example database itself to topics related to the user, with the goal of providing a natural, personalized interaction for that specific user. To develop a more intelligent system, we could use additional information such as dialog histories and natural language analyses, but this is a problem to be resolved in a future study.

3. Method for Generating User-Adapted Examples

If example input phrases similar to the user’s utterances are included in the example database and the corresponding response examples are appropriate, an example-based spoken dialog system can provide natural responses. Thus, it is important to include example phrases likely to be employed by the user in the example database. Our system addresses this issue by generating user-adapted examples using profile information about the user. The framework of our adaptation method is shown in Fig. 3. The steps shown in this framework will be explained in the following sections.

3.1 Acquiring Words for User Adaptation

In this study, we call words related to the user which are acquired in order to generate user-adapted examples “words for user adaptation”. First, information about the user is retrieved from the Web using basic information about the user in queries. Named entity extraction using KNP [19], a Japanese syntactic analyzer, and morpheme analysis using JUMAN++ [18], a Japanese morphological analyzer, are then applied to the retrieval results. Each named entity in the user’s profile data is classified using IREX (Information Retrieval and Extraction eXercise). Table 1 shows an example of named entity classifications and examples of specific details. In this study, we acquired words which fall into the categories of PERSON, LOCATION, ORGANIZA-
and ARTIFACT. But KNP often fails to detect the names of foods. For our task, the detection of food names is very important however, so we also used JUMAN++ and acquired the words which were labeled “artifact-food” as food-named entities. Thus, words acquired by named entity extraction and morpheme analysis are selected as candidates for “words for user adaptation”. Next, we calculate the TF-IDF (term frequency-inverse document frequency) score of the candidate words. TF-IDF is a numerical statistic that is intended to reflect the importance of a word within a document in a collection or corpus. TF represents the number of appearances in the search results, while IDF was calculated using Wikipedia data. In this study, the score for each word is obtained by normalizing the TF-IDF score for each search query. The following equation is used for normalization:

$$x_i^{\text{score}} = \frac{x_i - \mu}{\sigma}$$  \hspace{1cm} (1)

In Eq. (1), $x_i^{\text{score}}$ is the normalized value of word $i$ in word set $x$, $\mu$ is the average score and $\sigma$ represents the standard deviation of all of $x$. Words which exceed a threshold are considered to be related to the user and are selected as “words for user adaptation”. Figure 4 shows this procedure. Table 2 shows an example of user profiles. Table 3 shows examples of “words selected for user adaptation” which were acquired using the user’s profile when the TF-IDF score threshold was set to 0.6.

### 3.2 Adaptation of Examples

Example phrases are adapted to the user using the words acquired as described above. We generate user-adapted examples by calculating the similarity between words in the original examples and the acquired words using word2vec. Word2vec is a tool which reconstructs the linguistic contexts of words by using vectors to represent words in a 200-dimensional vector space. This distributed representation of words reveals their possible meanings based on word context. It also allows us to calculate the similarity between words mathematically using cosine similarity. We adapt the existing examples in a dialog example database to particular users by replacing words in the examples with the words acquired for user adaptation. Some original dialog examples are shown in Table 4. The procedure for generating user adapted examples is as follows. First, we acquire “words for user adaptation” as described above. Then we compare each of these words with the nouns contained in a dialog example database, which are identified through morphological analysis using MeCab[15]. Next, we calculate the sim-
Table 4  Samples of original dialog examples (Original Japanese sentences are written in parentheses.)

| Original Japanese | English |
|-------------------|---------|
| I want to go abroad. (海外旅行がしたいです) | Where do you want to go? (どこに行きたいですか) |
| I want to see an aurora. (オーロラが見たい) | I want to see that, too. (私も見たいです) |
| I like winter. (冬が好きだよ) | That’s when Christmas and New Year are. (クリスマスや正月がありますね) |
| I came from Okinawa. (沖縄出身です) | That’s where Shuri castle is. (那覇城があります) |
| I want something sweet. (甘いものが欲しい) | I want to eat something like. (ケーキとか食べたいですね) |

Fig. 5  System response selection in an example-based dialog system

Fig. 6  Generation of user-adapted examples using vector operation

4. Generation of User-Adapted Examples Using Arithmetic Operations

By replacing a word in a dialog example, the user-adapted example sometimes does not make sense. For example, when “Tokyo” in the sentence “Meiji Shrine in Tokyo is famous” is replaced with the user-adapted word “Kyoto”, the new sentence, “Meiji Shrine in Kyoto is famous”, is inconsistent, because that shrine is not located in Kyoto. “Meiji Shrine” should be replaced by the name of a famous shrine located in Kyoto, such as “Shimogamo Shrine”. To solve this problem, we propose applying an operation using distributed word representations, explained below, when the system finds another named entity in the example after replacing a word.

Distributed representations of words are generated using word2vec, enabling us to use vectors to calculate similarities between words. Word2vec also has a feature which allows the application of addition and subtraction operations to word meanings. For example, the distributed representation obtained from the calculation Tokyo − Japan + France is Paris. Our proposed system can generate user-adapted examples using this feature by replacing one of the words in a standard dialog example associated with the name of a place or a facility with a user-adapted word. If the new dialog example does not make sense, we can replace another word in the example using addition and subtraction, in order to generate an appropriate user-adapted example. For the example cited at the top of this section, where “Tokyo” is replaced with “Kyoto”, we can use this property to obtain the name of famous shrines in Kyoto to replace “Meiji Shrine”. We perform the calculation Meiji Shrine − Tokyo + Kyoto to obtain the names of shrines in Kyoto. Words located near the resulting vector then become potential candidates. In this study, we focus on specific words as described in Sect. 3.1. Figure 6 shows this procedure. Other researchers have actually used this arithmetic operation for some natural language processing. For example, Mitra [16] adopted this operation to web search query prediction. The system performs the completion of web queries using the inputted part of the query along with an operation on the distributed word representations of previous queries. However, to our knowledge, our research is the first attempt to apply this technique to the generation of dialog examples.

One problem with this method is that the results of the addition and subtraction of distributed word representations using word2vec may produce incorrect results due to a lack of accuracy in the distributed word space. For example, in Table 5 we give the example Shimanto river −
Table 5  Erroneous examles generated when using the arithmetic operation (Shimanto river − Kochi + Tokushima), with scores before and after correction using word similarity.

| Original rank | Word               | Operation score | Similarity score | Corrected score (λ = 0.3) |
|---------------|--------------------|-----------------|------------------|---------------------------|
| 1             | Teshio-river       | 0.750           | 0.402            | 0.646                     |
| 2             | Yoshino-river      | 0.747           | 0.548            | 0.687                     |
| 3             | Naka-river         | 0.743           | 0.473            | 0.662                     |

Fig. 7  Select a replacement word to generate the user-adapted example

Kochi + Tokushima. The correct word is “Yoshino-river” (Yoshinogawa), but as shown in the table, Yoshino-river is not the top ranked responseootnote{Improving the accuracy of word2vec by training it with huge amounts of text data would improve the accuracy of the arithmetic operation results, but this is outside the scope of this present study.}. For this reason, it is necessary to select the correct word from a list of candidate words obtained from the calculation results. The correct word is selected using two values, similarity with the result of the addition and subtraction operation and similarity with the word for user adaptation. First, we obtained multiple candidates whose distributed representation is similar to the results of the addition and subtraction operation, and then we compared the scores of each word obtained using the following formula to rescore them:

\[
\text{score} = \lambda \cdot \text{similarity}_1(w) + (1 - \lambda) \cdot \text{similarity}_2(w) \quad (2)
\]

In Eq. (2), \(\text{similarity}_1(w)\) represents cosine similarity between \(w\) and the result of the addition and subtraction of replacement words, while \(\text{similarity}_2(w)\) represents cosine similarity between \(w\) and the word for user adaptation. When only using the first term some incorrect words are selected accidentally. The second term is expected to filter out these incorrect words. Using this filtering method, the correct word can sometimes obtain the top similarity score, as shown in the fifth column of Table 5. In this study, we set \(\lambda\) to 0.3, and selected the word with the highest Eq. (2) score, and then generated adapted examples through replacement. Figure 7 shows this procedure.

5. Experimental Evaluation

5.1 Experimental Conditions

We used MeCab-ipadic-NEologd\cite{17} as the morpheme dictionary, since this dictionary is more robust when encountering new words than the conventional IPA Dictionary, and there was no need to perform consolidation of noun phrases into compound words. This allows our system to better analyze proper nouns such as the names of buildings and place names. In addition, we obtained semantic information about words using JUMAN++\cite{18} in order to select the words to be used for addition and subtraction in distributed representations.

We retrieved information related to a user’s profile using Google. Each named entity in the user’s profile was input into the Google Search API, one by one. Only location names were input after the user’s profile had been expanded by adding the word “tourist spot” (観光地) in Japanese. Then we applied named entity extraction using KNP\cite{22} to the retrieved web texts and selected the named entities which obtained normalized TF-IDF scores higher than 0.6 for user adaptation. Word2vec was used to calculate similarity and was trained using the Japanese version of Wikipedia dated July 1, 2017. The model of word2vec used was Skip-Gram, which takes context into consideration using 5-words on either side of the target word, and obtained 200-dimensional word vectors. Three sets of adapted examples were generated using the profiles of three people and one set of original dialog examples which contained 100 manually created examples.

To evaluate the performance of our proposed method, we examined the user-adapted examples which were generated using each set of dialog examples and manually evaluated them to be natural or unnatural.

5.2 Experimental Results

We called the total number of user-adapted dialog examples generated in our experiments “#Generated”, while the total number of generated examples which were determined to be natural was labelled “#Success”. We calculated our success rate using the following formula:

\[
\text{Success rate}(\%) = \frac{\#\text{Success}}{\#\text{Generated}} \times 100. \quad (3)
\]

5.2.1 Adaptation by Replacement of One Word in Each Example

First, we replaced just one word in each dialog example, using only the calculated word2vec similarities. We also changed the threshold for cosine similarity between the word in the example and a word for adaptation. The threshold was used to decide which replacement words were selected. Table 6 shows #Generated, #Success and Success Rate when using different thresholds. When we raised the threshold, the #Generated, #Success and Success Rate decreased because many of the examples become inconsistent. These inconsistent examples required addition and subtraction operations, and we had to replace more than one word,
Table 10 Example of adaptation generation using vector addition and subtraction operations with manually generated examples

| Original dialog example | User: Tokyo is a nice place.          |
|------------------------|--------------------------------------|
| Words for user adaptation | Osaka                                  |
| Replacement of similar words | Tokyo → Osaka                        |
| Calculation | Akihabara – Tokyo + Osaka |
| Replacement using above calculation | Akihabara → Umeda |
| Generated dialog example | User: Osaka is a nice place.           |
|                        | Sys: Have you been to Umeda?         |

Table 6 Experimental results for selected thresholds

| Similarity threshold | 0.70 | 0.73 | 0.75 | 0.78 |
|----------------------|------|------|------|------|
| #Generated           | 806  | 551  | 394  | 208  |
| #Success             | 168  | 101  | 70   | 36   |
| Success Rate         | 20.8 | 18.3 | 17.7 | 17.3 |

Table 7 Experimental results for each user profile with a threshold of 0.73

|       | User A | User B | User C | Total |
|-------|--------|--------|--------|-------|
| #Generated | 122    | 89     | 340    | 551   |
| #Success   | 37     | 33     | 31     | 101   |
| Success Rate | 30.4   | 37.0   | 9.1    | 18.3  |

Table 8 Experimental results using addition and subtraction on manually generated examples with a threshold of 0.73

|       | User A | User B | User C | Total |
|-------|--------|--------|--------|-------|
| #Generated | 122    | 89     | 340    | 551   |
| #Success   | 44     | 40     | 81     | 165   |
| Success Rate | 36.0   | 44.9   | 23.8   | 31.9  |

Table 9 Breakdown of addition and subtraction operation results

|                        | #Generated | #Addition and Subtraction Operations | #Success |
|------------------------|------------|-------------------------------------|----------|
|                        | 551        | 311                                 | 81       |

5.2.2 Adaptation Using Arithmetic Operation on the Distributed Representations

We then generated examples using addition and subtraction on the distributed representations, setting the threshold for cosine similarity to 0.73. The replacement word candidates derived using addition and subtraction had similarity scores of 0.6 or higher to the original words. In other words, we set the threshold for Eq. (2) to 0.6. We performed adaptation on examples with a $\lambda$ of 0.3 in Eq. (2). Table 8 shows our experimental results when using addition and subtraction, and Table 9 shows a breakdown of experimental result using addition and subtraction. Note that our system first generates adapted examples by replacing one word in each example. Then using the arithmetic operations on the distributed word representations, it modifies the adapted examples by replacing one more word. Thus, the rows of #Generated in Table 7 and Table 8 are identical.

A total of 311 dialog examples were modified using addition and subtraction, out of the original 551 examples. The number of natural examples generated was 81. Note that the set consisting of these 81 natural examples differs from the set consisting of 101 initially successful examples in Table 7 (i.e., these sets are disjoint), because the originally correct examples became incorrect after applying the adaptation operations, i.e., some of the successfully generated examples shown in Table 7 became unnatural due to the addition and subtraction operation, but overall, we were able to generate 64 additional natural dialog examples compared to when we created the examples without using addition and subtraction operations. From Tables 7 and 8, we can see that by using the addition and subtraction operation on the distributed word representations, #Success increased by 64 (from 101 to 165). In Table 9, #Success using this operation was 81. Therefore, 17 (= 81 – 64) adapted examples using the operation were incorrect. Another 213 (= 311 – 81 – 17) examples were also modified using the operation, but the good examples remained good, while the bad examples remained bad. Table 10 shows user-adapted examples when using addition and subtraction on distributed word representations. Akihabara is a famous district in Tokyo, but the user’s profile indicated he or she lives in Osaka. First, the system replaced Tokyo with Osaka, and then Akihabara was replaced with Umeda, a district in Osaka. As a result, we obtained a new, consistent and natural example.

6. Conclusion

In this study we developed a user adaptation technique for creating personalized sets of input and output dialog examples using user profile information, in order to build a dialog system that could chat naturally with a particular
user. We generated user-adapted examples by calculating the similarity between acquired words related to the user and the original words in the dialog examples. We then replaced words which were grammatically the same (nouns and proper nouns), whose similarity scores were higher than a threshold. As the final step of the adaptation process, we applied addition and subtraction operations to the distributed word representations, in order to generate additional natural examples.

In our evaluation experiment, we applied our method to selected dialog examples using information from the profiles of three people. We were able to generate natural user-adapted dialog examples through replacement based on distributed representations and the use of mathematical operations. By selecting replacement words from these candidates, we were able to improve our overall success rate to 31.9%.

However, in our experiment we observed the creation of many inappropriate examples as a result of the application of addition and subtraction operations. Table 11 shows an example of such a failure. “Osaka” should not have been replaced with “Fukui”, but the replacement decision is a very difficult one because it requires actually understanding what is going on in the original example. This more difficult problem remains to be resolved in the future. Other future work includes increasing our success rate by improving the accuracy of the addition and subtraction operations on word representations. We also have to think about methods we can use to replace more than two words in an example.

Acknowledgements

This study was partially supported by the Strategic Information and Communications R&D Promotion Program (SCOPE) of the Ministry of Internal Affairs and Communications of Japan and the JSPS KAKENHI Grant-in-Aid for Scientific Research, Grant Numbers 19H01125 and 19K04311.

References

[1] S. Ashida, “The effect of reminiscence music therapy sessions on changes in depressive symptoms in elderly persons with dementia,” J. Music Ther., vol.37, no.3, pp.170–182, 2000.
[2] C. Lee, S. Jung, S. Kim, and G.G. Lee, “Example-based dialog modeling for practical multi-domain dialog system,” Speech Commun., vol.51, no.5, pp.466–484, 2009.
[3] S. Han, K. Lee, D. Lee, and G.G. Lee, “Counseling dialog system with 5w1h extraction,” Proc. SIGDIAL 2013 Conference, pp.349–353, 2013.
[4] K. Lee, P.H. Seo, J. Choi, S. Koo, and G.G. Lee, “Conversational Knowledge Teaching Agent that Uses a Knowledge Base,” Proc. 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp.139–143, 2015.
[5] T. Mikolov, W.-T. Yih, and G. Zweig, “Linguistic regularities in continuous space word representations,” Proc. 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.746–751, 2013.
[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv:1301.3781, 2013.
[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” Advances in Neural Information Processing Systems, pp.3111–3119, 2013.
[8] Y. Goldberg and O. Levy, “Word2vec explained: Deriving Mikolov et al.’s negative-sampling word-embedding method,” arXiv:1402.3722, 2014.
[9] X. Rong, “Word2vec parameter learning explained,” arXiv:1411.2738, 2014.
[10] R. Nishimura, Y. Nishihara, R. Tsurumi, A. Lee, H. Saruwatari, and K. Shikano, “Deploying a speech-based information system as a research platform for speech recognition research in real environments,” Electronics and Communications in Japan (Part II: Electronics), vol.88, no.12, pp.43–54, 2005.
[11] R. Nishimura, A. Lee, H. Saruwatari, and K. Shikano “Public speech-oriented guidance system with adult and child discrimination capability,” 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, pp.1–433-6, 2004.
[12] T. Cincarek, H. Kawanami, R. Nishimura, A. Lee, H. Saruwatari, K. Shikano, “Development, long-term operation and portability of a real-environment speech-oriented guidance system,” IEICE Trans. Inf. & Syst., vol.91, no.3, pp.576–587, 2008.
[13] C. Lee, S. Jung, and G.G. Lee, “Robust dialog management with n-best hypotheses using dialog examples and agenda,” Proc. ACL-08: HLT, pp.630–637, 2008.
[14] K. Kim, C. Lee, S. Jung, and G.G. Lee, “A frame-based probabilistic framework for spoken dialog management using dialog examples,” Proc. 9th SIGdial Workshop on Discourse and Dialogue of the Association for Computational Linguistics, pp.120–127, 2008.
[15] T. Kudo, K. Yamamoto, and Y. Matsumoto, “Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing,” 2004.
[16] B. Mitra, “Exploring Session Context using Distributed Representations of Queries and Reformulations,” Proc. 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR ’15, pp.3–12, 2015.
[17] T. Sato, “Mecab-ipadic-NEologd: neologism dictionary for MeCab,” https://github.com/neologd/mecab-ipadic-neologd, 2016.
[18] H. Morita, D. Kawahara, and S. Kurohashi, “Morphological analysis for unsegmented languages using recurrent neural network language model,” Proc. 2015 Conference on Empirical Methods in Natural Language Processing, pp.2292–2297, 2015.
[19] D. Kawahara and S. Kurohashi, “A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis,” Proc. main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, pp.176–183, 2006.

[20] P.-J. Chen, I.-H. Hsu, Y.-Y. Huang, and H.-Y. Lee, “Mitigating the impact of speech recognition errors on chatbot using sequence-to-sequence model,” 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp.497–503, 2017.

[21] K. Yu, Z. Zhao, X. Wu, H. Lin, and X. Liu, “Rich short text conversation using semantic-key-controlled sequence generation,” IEEE/ACM Trans. Audio, Speech, Language Process., vol.26, no.8, pp.1359–1368, 2018.

[22] D. Kawahara and S. Kurohashi, “Fully-Lexicalized Probabilistic Model for Japanese Syntactic and Case Structure Analysis,” Proc. HLT-NAACL 2006, pp.176–183, 2006.

[23] E. Seto, R. Nishimura, and N. Kitaoka, “Customization of an example-based dialog system with user data and distributed word representations,” Proc. APSIPA 2018, pp.1718–1724, 2018.

Norihide Kitaoka received his B.S. and M.S. degrees from Kyoto University, Japan. In 1994, he joined DENSO CORPORATION. In 2000, he received his Ph.D. degree from Toyohashi University of Technology (TUT), Japan. He joined TUT as a research associate in 2001 and was a lecturer from 2003 to 2006. He was an associate professor at Nagoya University, Japan, from 2006 to 2014 and joined Tokushima University, Japan, as a professor in 2014. He is now a professor in TUT from 2018. His research interests include speech processing, speech recognition, and spoken dialog systems. He is a member of IEEE, International Speech Communication Association (ISCA), Asia Pacific Signal and Information Processing Association (APSIPA), Information Processing Society of Japan (IPSJ), Acoustical Society of Japan (ASJ), The Japanese Society for Artificial Intelligence (JSAI), and The Association for Natural Language Processing.

Eichi Seto received his B.S. and M.S. degree from Tokushima University, Japan. He was engaged in the study on spoken dialog systems. He currently works for DENSO TEN Limited.

Ryota Nishimura received his B.S., M.S. and Ph.D. degree from Toyohashi University of Technology (TUT), Japan. He joined TUT as a researcher in 2011. He was a researcher of Nagoya University, Japan, from 2011 to 2012. He was an assistant professor of Nagoya Institute of Technology, Japan, from 2012 to 2015. He was an assistant professor of Keio University, Japan, from 2015 to 2017. He was a researcher of Tokushima University, Japan, from 2017 to 2018, and is an associate professor from 2018. His research interests include spoken dialog system and spoken language information processing. He is a senior member of IEEE, and a member of IPSI, ASJ, JSAI, and Phonetic Society of Japan (PSJ).