# Improving the Performance of Chat-oriented Dialogue Systems via Dialogue Breakdown Detection

Michimasa Inaba, Kenichi Takahashi

**Abstract** Dialogue breakdown detection is a technique used for identifying inappropriate utterances in dialogue systems that has attracted increased attention, especially in chat-oriented dialogue systems. Although it is generally assumed that dialogue breakdown detection avoids generating system responses that then cause difficulties in continuing the given dialogue, this has yet to be verified experimentally or theoretically. In this paper, we apply the dialogue breakdown detection technique to generate responses for a chat-oriented dialogue system and experimentally verify that performance is improved by measuring the extent to which dialogue breakdown is avoided. Our experimental results show that dialogue breakdown detection indeed is able to improve the appropriateness of system responses; however, short, simple, and dull responses tend to increase when using this technique.

## 1 Introduction

Dialogue breakdown detection is a technique for identifying inappropriate utterances in dialogue systems, especially in chat-oriented dialogue systems. International competitions focused on the use of this technique are held annually, in particular the dialogue breakdown-detection challenge (DBDC) that has been held every year since 2015 [5, 3, 4].

Although it is generally assumed that dialogue breakdown detection avoids generating system responses that then cause difficulties in continuing the given dialogue, this has yet to be verified experimentally or theoretically. Further, specific application methods for using this technology remain unclear, in particular in how it can specifically improve the performance of dialogue systems.

Michimasa Inaba, Kenichi Takahashi
Hiroshima City University, Japan, e-mail: {inaba, takahashi}@hiroshima-cu.ac.jp

Given the above, in this paper, we apply the dialogue breakdown detection technique to chat-oriented dialogue systems and experimentally verify that the performance is indeed improved by successfully avoiding breakdowns in dialogue.

## 2 Related Works

In [11], Sugiyama proposed a method for applying it to example-based dialogue systems. More specifically, this method searches utterances similar to input (i.e., user) utterances in a dialogue database using word vectors provided by Word2Vec [7]; next, this method re-ranks the top 20 response candidates using breakdown probabilities provided by the dialogue breakdown detection method, then uses these top-ranked utterances as its responses. In [8], Mori and Araki proposed a dialogue system using breakdown probabilities as a criterion for selecting responses from among multiple candidate utterance-generation modules; however, comparative evaluations without using dialogue breakdown detection was not conducted in these studies, and there was no evaluation of how effective the application method was in improving system performance.

## 3 Dialogue Systems

To properly analyze alterations in responses and the performance of systems using breakdown detection, we decided to focus on systems that generate multiple response candidates for each given input. Therefore, in this study, we do not cover typical rule-based dialogue systems, such as ELIZA [14] or ALICE [13] in which responses to inputs are fixed, because such systems cannot produce alternative candidates if the generated response is then estimated as potentially breaking down the given dialogue.

Systems that do generate multiple responses often score their candidates and rank them. In this study, we re-rank candidates by altering the given scores or ranks by applying dialogue breakdown detection, then analyze changes in performance.

For our analysis, we used the following three types of Japanese chat-oriented dialogue systems.

### Example-based System (IRS)

The example-based system that we used was an example-based dialogue system based on IR-STATUS [9] also used in DBDC2 held in 2016. This system uses Apache Lucene for its search engine and consists of 26,972 input-response pairs extracted from human dialogues as its example database. Other parameters and settings of the Apache Lucene were set to the various defaults.

To apply dialogue breakdown detection to this system, we used the top 10 matching utterances as response candidates and the similarity functionality inherent in Apache Lucene to calculate the corresponding response scores.

N̲eural Conversational Model (NCM)

The neural conversational model (NCM) system that we used was a dialogue system with an encoder-decoder neural network for response generation. We used KyotoNMT [1] for implementing this system. The encoder and decoder components were four-layer long short-term memory (LSTM) models, each with 1000 hidden cells. The vocabulary size was limited to 80,000 terms, and the dropout rate was set to 0.2. Further, the model was trained using 10 million tweet-reply pairs with an Adam optimizer.

In our implementation, response candidates were generated using 12 models individually trained with different initial parameters. Note that when using the encoder-decoder neural network, multiple responses can be generated using beam search; however, since our one trained model generated almost the same response candidates (e.g., as present and future tenses), we decided to use multiple models. For response scores, we used the average word probability of each response; further, if more than one model generated the same response, the highest score was used as the response score. Therefore, when the system chats with a human subject for data acquisition (as we discuss in Section 4), it only uses one model to reduce response times.

N̲eural Utterance Ranking Model (NUR)

The neural utterance ranking (NUR) system that we used was based on our neural network-based dialogue model [6]; this model ranks candidate utterances acquired from Twitter in response to the given dialogue context and uses the highest-ranked candidate as its response. The training settings and data set that we used in our present study were the same as those described in [6]; further, we used response scores for utterance ranking in our NUR model for the system response scores in our present study.

## 4 Dialogue Breakdown Detection

In this study, the dialogue format and various settings regarding dialogue breakdown detection are based on those of DBDC2. Here, dialogue data consists of text chats between a dialogue system and a user in Japanese; annotations are also included for each system response, where annotations are one of the following three breakdown labels:

(NB) Not a breakdown  It is easy to continue the conversation.
(PB) Possible breakdown  It is difficult to continue the conversation smoothly.
(B) Breakdown  It is difficult to continue the conversation.

For each system utterance, annotations were provided by more than 30 individuals. Next, we applied a detection method to estimate majority labels among all annotators for each utterance, as well as a distribution of breakdown labels.

### *4.1 Methodology*

The current state-of-the-art dialogue breakdown detection method described in [12] was proposed in DBDC3; however, this method is difficult to implement because it utilizes some feature sets based on an unpublished corpus with original annotations.

In this study, we use the best performance detection method from DBDC2 [10]; this method uses typical error patterns of system responses, such as abrupt changes in the discussed topic or unnatural connections of dialogue acts as features, then estimates a distribution of labels using the extra-trees regressor [2].

As a preliminary experiment, we compared cases in which all system data were used as training data for all systems with cases in which data was divided for each system and learned individually. Our initial results indicated that the individual case was better than the all-in-one case; therefore, the breakdown detection model we used for the remainder of our study was the approach in which each system learns individually.

### *4.2 Dataset*

As training data for the breakdown detection method, we used dialogue data from the three types of systems — i.e., IRS, NCM, and NUR — with annotations accompanying the dialogue breakdown labels. We collected dialogue data for the NCM and NUR systems in full compliance with DBDC2 data collection and annotation rules. Since the IRS system data have already been collected in DBDC2, we simply use this data as IRS data for our present study.

Resulting statistics for the above data are summarized in Table 1, indicating that the NUR system generated the fewest number of responses that caused a dialogue breakdown, and the IRS system was the opposite, based on the ratio of PB and B labels.

**Table 1** Statistics of Data for Dialogue Breakdown Detection

| | IRS | NCM | NUR |
|---|---|---|---|
| Number of dialogues | 100 | 100 | 120 |
| Number of user utterances | 1000 | 1000 | 1200 |
| Number of system utterances | 1100 | 1100 | 1320 |
| Number of annotators | 30 | 30 | 34 |
| NB (Not a breakdown) | 31.1% | 47.4% | 57.7% |
| PB (Possible breakdown) | 26.7% | 32.7% | 27.0% |
| B (Breakdown) | 42.1% | 17.2% | 15.2% |
| Fleiss' $\kappa$ | 0.29 | 0.29 | 0.26 |
| Fleiss' $\kappa$ (PB+B)[1] | 0.38 | 0.43 | 0.42 |

**Table 2** Statistics for Response Candidate Re-ranking

| | IRS | NCM | NUR |
|---|---|---|---|
| Number of data points (Number of context and response pairs) | 300 | 300 | 300 |
| Number of utterances in context | 1.37 | 2.14 | 2.04 |
| Number of candidate responses per data | 11.42 | 10.57 | 10.94 |
| Number of words per candidate | 18.13 | 9.54 | 10.70 |
| Number of human annotators | 4.61 | 5.92 | 3.88 |

## 5 Experiments

Our experiments consisted of re-ranking candidate responses generated by the given systems using the results of our dialogue breakdown detection method.

To evaluate the response performance of each system after applying breakdown detection, we constructed a data set that included all context/response pairs, i.e., the chat log between two speakers as context and 10 or more response candidates generated by the given system; note that these pairs also incorporated scores and breakdown labels.

Each response candidate is then annotated using breakdown labels via the same method described in Section 4.2 above by at least three annotators. In evaluating response performance, we regard candidates with 50% or more annotators decided as NB as correct response and others as incorrect. Note that the data collection procedure described here is the same as that used in [6], and test data from [6] is therefore used for the NUR system in our experiments. The contexts of the data for the IRS and NCM systems were acquired from the data set used in [6] with the condition that a system must generate 10 or more response candidates. Statistics of the data are summarized in Table 2.

[1] Fleiss' $\kappa$ when PB and B are treated as a single label.

### *5.1 Re-ranking Method*

To apply dialogue breakdown detection, we propose the three response re-ranking methods described below.

**Classification-based method** This method focuses on the classification results output by the breakdown detection method, lowering the ranking of response candidates classified as causing a dialogue breakdown. Here, labels with the maximum probability according to the detection model are judged as estimated results; further, we assume that only the B label is a breakdown, and therefore lump PB into the B category.

**Probability-based method** This method re-ranks candidates by calculating the product of the response score and the non-breakdown probability, using this product as the new score. From estimated probabilities $p(NB)$, $p(PB)$ and $p(B)$, we assume that $p(NB)$ is the non-breakdown probability and $p(NB) + p(PB)$ is a non-breakdown probability.

**Regression-based method** Using estimated probabilities $p(NB)$, $p(PB)$, $p(B)$ and the response score $s$ as input features for a linear regression model, candidates are re-ranked with new score $s_{new}$ as calculated by the model. For training the regression model, we used the mean squared error of $s_{new}$ with the teacher score of a correct candidate as 1.0 and that of an incorrect candidate as 0.0 as the loss function. This regression-based method includes the optimization of parameters, with evaluation performed using 10-fold cross-validation.

### *5.2 Results*

To evaluate re-ranking performance, we used the mean average precision (MAP) measure, which indicates how many correct response candidates ranked higher. Fig. 1, 2 and 3 show MAP results for the top $n$ ranked candidate utterances.

Based on this MAP measure, probability-based (NB), probability-based (NB + PB), and regression-based methods all showed performance improvements across all systems as compared to results without re-ranking. For the classification-based (B) and (PB+B) cases, the increased effectiveness differed from system to system. Individually, for the IRS results (i.e., Fig. 11), the probability-based (NB) method was the most effective, followed by the probability-based (NB+PB) method, then the regression-based method. The classification-based (B) and (PB+B) cases had very similar results, with smaller improvements versus that of other methods, but very large improvements were observed when compared to those without re-ranking. For the NCM results (i.e., Fig. 2), the observed improvement was larger for the regression-based method, then for the probability-based (NB) method, followed by the probability-based (NB+PB) method. Conversely, the classification-

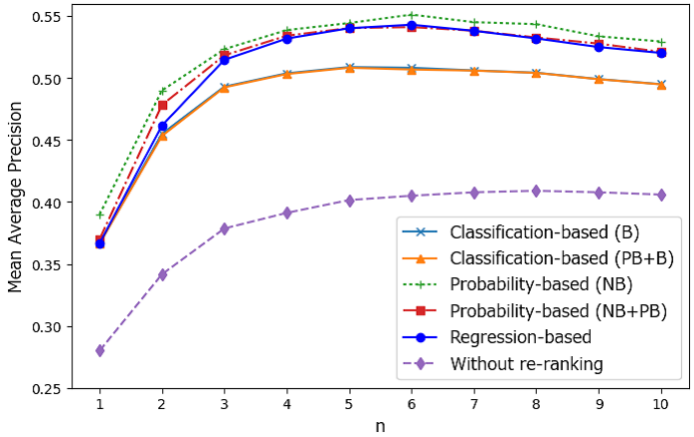


**Fig. 1** MAP over top n response candidates (IRS)

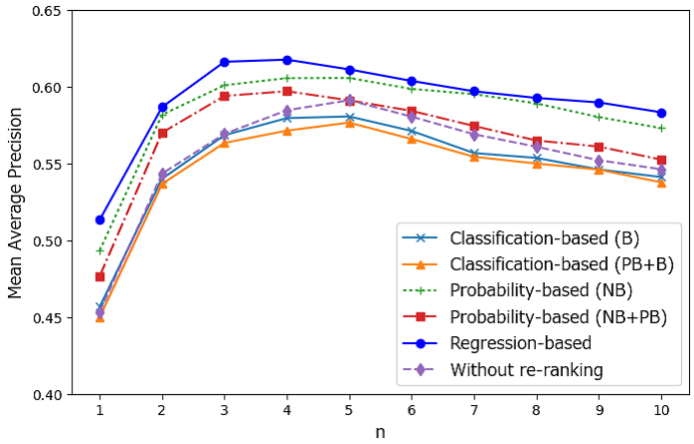


**Fig. 2** MAP over top n response candidates (NCM)

based (B) and (PB+B) methods showed deteriorated performance as compared to results without re-ranking. Finally, for the NUR results (i.e., Fig. 3), the regression-based method showed the best performance improvement, but improvements were smaller versus those of the other systems.

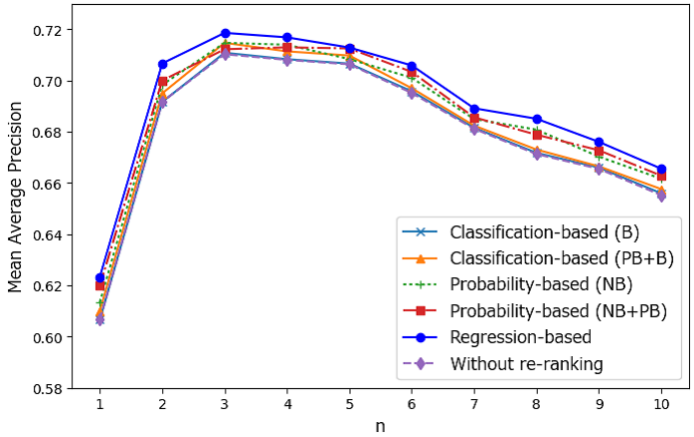


**Fig. 3** MAP over top n response candidates (NUR)

## *5.3 Discussion*

### 5.3.1 Distribution of labels and performance improvements

Through our work, we confirmed that response performance can be improved by applying dialogue breakdown detection; however, the effective re-ranking method and degree of effectiveness here varied depending on the system. To investigate the cause, we further analyzed the output of the dialogue breakdown detection method.

Table 3 shows the distribution of labels with maximum probability output by the breakdown detection method. From the table, we observe that the ratio of the B label is much larger than the others for IRS. Therefore, in the case of IRS, since more scores are changed by the detection method, the range of performance improvement increased accordingly. For the NCM and NUR methods, since more than 80 % of the responses are NB labels, the range of performance improvements became very small. In particular, since most labels were NB in NUR, the classification-based method using only the PB and B labels was less effective, whereas the probability-based and regression-based methods using NB label information were more effective.

**Table 3** Distribution of labels output by the dialogue breakdown detection method. Numbers in parentheses indicate the number of labels.

| | NB (Not a breakdown) | PB (Possible breakdown) | B (Breakdown) |
|---|---|---|---|
| IRS | 20.7% (711) | 0.3% (10) | 79.0% (2706) |
| NCM | 84.9% (2862) | 9.4% (317) | 5.7% (193) |
| NUR | 96.6% (3173) | 2.1% (69) | 1.3% (42) |

### 5.3.2 Performance deterioration in NCM

In NCM, the classification-based (B) and (PB + B) methods resulted in worse performance versus performance without re-ranking. We investigated the ratio of correct responses among the response candidates with the B label, finding that IRS was 11.6 % and NUR was 19.0 %, whereas NCM was 39.4 %, more than twice the other values. Therefore, the performance of breakdown label classification was poor for NCM, which likely caused response performance to deteriorate.

### 5.3.3 Dialogue breakdown detection and re-ranking performance

We first note that it is important to improve the performance of dialogue breakdown detection to improve response performance of dialogue systems. To analyze response performance when breakdown detection performance is low, we investigated re-ranking performance by reducing the training data set size of the detection model. Fig. 4 shows MAP results using the top-ranked response (i.e., MAP@1) while increasing the size of the learning data set by steps of 10. Here, we used the probability-based (NB) method for re-ranking. From the figure, we observe that MAP@1 generally improves as the size of the training data set increases, though the NUR results are only slightly improved. Therefore, we were able to confirm that it is important to improve the performance of the dialogue breakdown detection model before applying its results to dialogue systems.

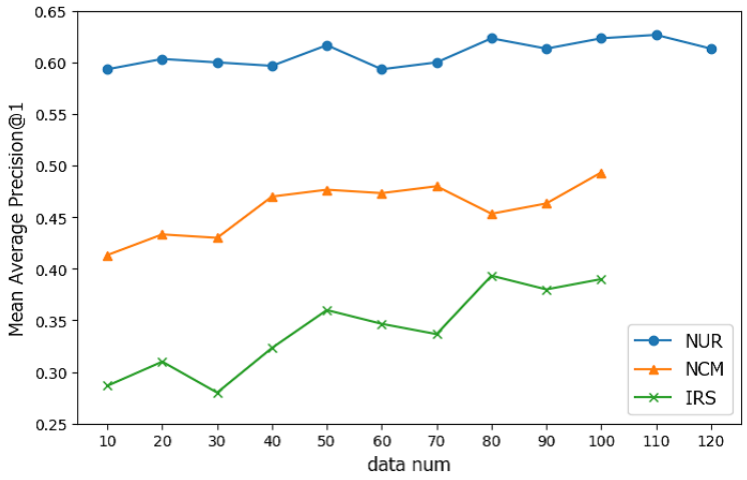


**Fig. 4** Training data size and response performance

### 5.3.4 Changes in response

To analyze the change in response in the case of re-ranking, we calculated the number of tokens and the number of types included in the top-ranked responses when both applying and not applying our breakdown detection method. Here, we used the regression-based method for the re-ranking method.

We present our results in Table 4; as shown in the table, we found that all systems tended to decrease both the number of tokens and the number of types after applying dialogue breakdown detection. In particular, the amount of change that occurred in IRS was rather large.

**Table 4** Changes in the Number of Tokens and Types of Responses with and without Re-ranking

| | Number of tokens (without re-ranking) | Number of tokens (re-ranking) | Number of types (without re-ranking) | Number of types (re-ranking) |
|---|---|---|---|---|
| IRS | 18964 | 12276 | 1219 | 761 |
| NCM | 11143 | 9317 | 363 | 332 |
| NUR | 10659 | 10494 | 885 | 881 |

Tables 5, 6 and 7 present examples of response changes after applying the regression-based method. More specifically, in Tables 5 and 6, responses generated after re-ranking tended to change into shorter and simpler responses. Unfortunately, we note here that if a system often generated such short and simple responses, while it is possible to avoid dialogue breakdown, the risk increases for boring the user because the system does not provide any new or interesting information. We therefore conclude here that it is necessary to pay close attention so that only such responses will not be chosen when the dialogue breakdown detection is applied to a system in which a simple response is included in the response candidates.

**Table 5** Change in Responses after Applying Dialogue Breakdown Detection (IRS)

| Context | Without Re-ranking | Re-ranking |
|---|---|---|
| A: Russian coke tastes good! (ロシアンコーラうますぎィ！) | Violin! It is a waste that you practiced for 8 years but you cannot play now. Don't you begin again? (バイオリン！８年やって今弾けないってもったいないですね。再開はしないんですか？) | That's delicious. (うまいね) |
| A: I'd like to buy a file and a card reader tomorrow, but it is hassle. (明日カードリーダーとファイル買いたいんだけどめんどい) | It would be nice to have that. I will devote. (そうなってるといいですね〜。精進します。) | Me too. lol (私もです。笑) |
| A: The timing is bad now. (ダメだ今はタイミングが悪い)<br>B: Are you OK? (大丈夫？)<br>A: No. (だめ) | I do not think that it will be raised so firmly. (そこまでしっかりと育てられるとは思いません。) | OK. Besides, it is not bad. (いいですよ。それに、だめじゃないですよ。) |

**Table 6** Change in Responses after Applying Dialogue Breakdown Detection (NCM)

| Context | Without Re-ranking | Re-ranking |
|---|---|---|
| A: I did not do my homework but I came to class. (課題やってないけど授業に来た) | That's right, you do not have any money. (そうなんだよね。お金がないんだよね。) | Really? (そうなんですか？) |
| A: I want to take a pillow in my laboratory (研究室に枕持ってこよ) | You do not bring it to the laboratory? (研究室に持ってきてないんですか？) | Please make me a pillow! (枕にしてください！) |
| A: Recently sleep time is increased (最近睡眠時間増えてやばい)<br>B: I wish I were you (羨ましいよ〜)<br>A: I'm dreaming of a ghost coming out (幽霊が出てくる夢を見るのだが) | Is it close from Kamata? (蒲田から近いんだっけ？) | Really (そうなんですか) |

**Table 7** Change in Responses after Applying Dialogue Breakdown Detection (NUR)

| Context | Without Re-ranking | Re-ranking |
|---|---|---|
| A: Let's calm down by saying that you can do today can be done tomorrow! (今日できることは 明日でもできるってそう言ってこころを落ち着かせよう！) | The mind and body are different (こころと体は別だよね) | Maybe I want to live with room in my mind (こころのゆとりが欲しいかも) |
| A:Let's do our best ... I have to work hard ... (I go to buy Red Bull) (もっと頑張ろう…頑張らなきゃ…… (レッドブルを買いに行く)) | Red Bull and Minmin-daha are not working well for me (レッドブルとか眠眠打破とかきかないんだよね) | Red Bull is the best energy drink (エナジードリンクはレッドブルが一番だね) |
| A: Recent Lexus' TV commercial is funny! (最近のレクサスの CM おもしろい！)<br>B: I don't like Lexus' face (レクサスお顔好きくないんよね)<br>A: How about Mazda's? (じゃあマツダはどう？) | I cannot say that Lexus is a fun car (レクサスは、楽しいクルマとは勿論言えない) | I am really looking forward to Mazda's sports concept car (マツダのスポーツコンセプトすごく楽しみだよ) |

## 6 Conclusions

In this present work, we experimentally evaluated whether response performance could be improved by applying a dialogue breakdown detection method to three different types of dialogue systems. We therefore proposed three types of response re-ranking methods for applying dialogue breakdown detection, i.e., a classification-based method that uses classification results of breakdown labels, a probability-based method that uses non-breakdown probability values, and a regression-based method that uses linear regression with the probability distribution of breakdown labels and response scores as the feature set. Our experimental results indicated that the probability-based and regression-based methods effectively improved per-

formance across all dialogue systems. Further, the classification-based method improved the response performance of two dialogue systems, but actually caused deteriorated performance for the third system. We also found that the number of short and simple responses increased when we applied dialogue breakdown detection as compared to results without such detection.

## Acknowledgements

This study received a grant of JSPS Grants-in-aid for Scientific Research 16H05880

## References

1. Cromieres, F.: Kyoto-nmt: a neural machine translation implementation in chainer. In: Proceedings of COLING 2016: System Demonstrations, pp. 307–311 (2016)
2. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine learning **63**(1), 3–42 (2006)
3. Higashinaka, R., Funakoshi, K., Inaba, M., Arase, Y., Tsunomori, Y.: The dialogue breakdown detection challenge 2. SIG-SLUD **B5**(02), 64–69 (2016)
4. Higashinaka, R., Funakoshi, K., Inaba, M., Tsunomori, Y., Takahashi, T., Kaji, N.: Overview of dialogue breakdown detection challenge 3. In: Proceedings of Dialog System Technology Challenge 6 (DSTC6) Workshop (2017)
5. Higashinaka, R., Funakoshi, K., Yuka, K., Inaba, M.: The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics. In: 10th edition of the Language Resources and Evaluation Conference (2016)
6. Inaba, M., Takahashi, K.: Neural utterance ranking model for conversational dialogue systems. In: 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 393–403 (2016)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013)
8. Mori, H., Araki, M.: Selection method of an appropriate response in chat-oriented dialogue systems. In: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 228–231 (2016)
9. Ritter, A., Cherry, C., Dolan, W.B.: Data-driven response generation in social media. In: Proceedings of the conference on empirical methods in natural language processing, pp. 583–593 (2011)
10. Sugiyama, H.: Chat-oriented dialogue breakdown detection based on the analysis of error patterns in utterance generation. SIG-SLUD **B5**(02), 81–84 (2016)
11. Sugiyama, H.: Utterance selection based on sentence similarities and dialogue breakdown detection on ntcir-12 stcv task. pp. 552–553 (2016)
12. Sugiyama, H.: Dialogue breakdown detection based on estimating appropriateness of topic transition. In: Dialog System Technology Challenges (DSTC6) (2017)
13. Wallace, R.: The anatomy of A.L.I.C.E. Parsing the Turing Test pp. 181–210 (2008)
14. Weizenbaum, J.: ELIZA-a computer program for the study of natural language communication between man and machine. Communications of the ACM **9**(1), 36–45 (1966)