The Dialogue Breakdown Detection Challenge: Task Description, Datasets, and Evaluation Metrics

Ryuichiro Higashinaka[1], **Kotaro Funakoshi**[2] **Yuka Kobayashi**[3], **Michimasa Inaba**[4]
[1]NTT Media Intelligence Laboratories
[2]Honda Research Institute Japan Co., Ltd.
[3]Toshiba Corporation
[4]Hiroshima City University

Abstract

Dialogue breakdown detection is a promising technique in dialogue systems. To promote the research and development of such a technique, we organized a dialogue breakdown detection challenge where the task is to detect a system's inappropriate utterances that lead to dialogue breakdowns in chat. This paper describes the design, datasets, and evaluation metrics for the challenge as well as the methods and results of the submitted runs of the participants.

Keywords: dialogue breakdown, chat-oriented dialogue, evaluation workshop

1. Introduction

Although voice agent services are beginning to appear on the market, the limited capabilities of these systems mean that humans and machines still cannot converse as naturally as two humans. The main problem is that systems typically make inappropriate utterances that lead to dialogue breakdowns. By dialogue breakdown, we mean a situation in a dialogue where users cannot proceed with the conversation (Martinovsky and Traum, 2003). To avoid this situation, technology for dialogue breakdown detection is essential because such technology will enable systems to avoid the creation of inappropriate utterances and also to identify dialogue breakdowns when they occur and perform the necessary recovery procedures.

This paper reports the results of our "Dialogue breakdown detection challenge", which is an evaluation workshop dedicated to dialogue breakdown detection. The aim of the challenge was to bring together various methods for dialogue breakdown detection and discuss possible evaluation metrics. Although there have been other dialogue-related evaluation campaigns, including a spoken dialogue challenge (Black and Eskenazi, 2009) and a dialogue state tracking challenge (Williams et al., 2013), our challenge is unique in that we deal with chat dialogues. We focus on chat because, compared to the errors in task-oriented dialogues (Bear et al., 1992; Carpenter et al., 2001; Bulyko et al., 2005), we can observe more diverse dialogue breakdown phenomena (Higashinaka et al., 2015a), which makes it possible to develop general and robust techniques for breakdown detection. Our challenge is also unique in that we deal with a highly subjective phenomenon, posing difficulties in creating appropriate evaluation metrics.

In this paper, we present how we designed the task, prepared the datasets, and conceived the evaluation metrics for the challenge. The event was held in October 2015 and there were six participants. We also report the detection methods submitted by the participants and provide their results. We finish with a discussion of the validity of the evaluation metrics we utilized.

2. Task description

The task of the dialogue breakdown detection challenge is to detect whether the system utterance causes dialogue breakdown in a given dialogue context. For example, consider the following piece of dialogue:

S1: It's nice to go shopping alone.
U1: I agree. That's nice.
S2: Shopping takes time.
U2: Window shopping is also fun.
S3: It's fun to go shopping with somebody.

Here, system utterances are prefixed by S and user utterances by U. The dialogue context is from S1 to U2 and the target utterance for dialogue breakdown detection is S3 (underlined). In this example, S3 is *likely* to be causing a dialogue breakdown because S3 contradicts S1. Therefore, a detector that classifies this as a dialogue breakdown will be regarded as accurate. The reason we say "likely" is that a human's decision on dialogue breakdown is highly subjective and it is difficult to decide on a single gold label. For this reason, we use many annotators for dialogue breakdown annotation and opt for majority voting and their probability distribution as references.

Given pairs of dialogue context and a succeeding system utterance, the participants of the challenge submit, for each pair, (1) a single dialogue breakdown label and (2) the probability distribution of the breakdown labels. Note that, although some utterances may exist after the target utterance, they cannot be used for prediction because, for this challenge, we focus on avoiding dialogue breakdown rather than recovery. In the challenge, each participant can submit up to three "runs", so several parameters for dialogue breakdown detection can be tested.

3. Datasets

We distributed two sets of data to participants: one consisting of training data and the other of development and test data. The training data are those that we previously made public as a "chat dialogue corpus"[1]. The development and

[1] https://sites.google.com/site/dialoguebreakdowndetection/

test data were newly created for this challenge (these data have also been made public on the same website).

3.1. Chat dialogue corpus

The chat dialogue corpus contains 1,146 text chat dialogues conducted between human users and a chat system. The language is Japanese. The users were recruited from among dialogue researchers and their collaborators. We used a chat system based on NTT Docomo's chat API[2], which is publicly available (technical details on the chat API can be found in (Onishi and Yoshimura, 2014)). Each dialogue contains 21 utterances (one system prompt followed by ten utterances each from the system and user in an alternate manner). See (Higashinaka et al., 2015a; Higashinaka et al., 2015b; Higashinaka et al., 2015c) for details on the data; the types of errors made by the system are also discussed in these studies. The chat dialogue corpus is divided into two parts: init100, which contains 100 dialogues with dialogue breakdown annotation by 24 annotators for each system utterance, and rest1046, which was annotated by two to three annotators. The following three breakdown labels were used:

(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.

Here, the labels were annotated depending on how easy/difficult it is to continue the conversation after each system utterance if the annotators were the dialogue participants in the dialogues in question; they did not predict if dialogue breakdown would actually happen or not in subsequent dialogues because it would be too difficult a task and would be similar to random guessing.
The statistics of the data are shown in Table 1. As indicated by Fleiss' κ, the breakdown annotation is highly subjective, which led to our decision to use majority voting and distribution-based evaluation metrics for evaluation (see the next section).

3.2. Development and test sets

For the challenge, we newly collected dialogue data and annotated them in the same manner as we collected the chat dialogue corpus, except that we used crowdsourcing (CrowdWorks[4] for dialogue collection and Yahoo! Crowdsourcing [5] for annotation). Here, each system utterance was annotated by 30 annotators. Fleiss' κ is lower than that for the chat dialogue corpus, probably because the dialogues were annotated by the general public rather than researchers, who have some notion of how the system may behave. The data were split into development (dev) and test (test) sets consisting of 20 and 80 dialogues, respectively. For the formal run, only the test set was used for evaluation.

4. Evaluation metrics

Since there are no established metrics for dialogue breakdown detection, we enumerated possible metrics. We created two types of evaluation metrics: classification-related and distribution-related.

4.1. Classification-related metrics

Classification-related metrics evaluate the accuracy related to the classification of the breakdown labels. Here, the accuracy is calculated by comparing the output of the detector and the gold label determined by majority voting. We use a threshold t to obtain the gold label: that is, we first find the majority label and check if the ratio of that label is above t. If so, the gold label becomes that label and NB otherwise. We used the following metrics.

- Accuracy: The number of correctly classified labels divided by the total number of labels to be classified.
- Precision, Recall, F-measure (B): The precision, recall, and F-measure for the classification of the B labels.
- Precision, Recall, F-measure (PB+B): The precision, recall, and F-measure for the classification of PB + B labels; that is, PB and B labels are treated as a single label.

These metrics can provide intuitive results about the detection of dialogue breakdowns because they directly evaluate whether dialogue breakdowns are correctly classified or not. However, the choice of an appropriate t value remains an open issue.

4.2. Distribution-related metrics

Distribution-related metrics evaluate the similarity of the distribution of the breakdown labels, which is calculated by comparing the predicted distribution of the labels with that of the gold labels. We used the following metrics.

- JS Divergence (NB,PB,B): Distance between the predicted distribution of the three labels and that of the gold labels calculated by Jensen-Shannon Divergence.
- JS Divergence (NB,PB+B): JS divergence when PB and B are regarded as a single label.
- JS Divergence (NB+PB,B): JS divergence when NB and PB are regarded as a single label.
- Mean Squared Error (NB,PB,B): Distance between the predicted distribution of the three labels and that of the gold labels calculated by mean squared error.
- Mean Squared Error (NB,PB+B): Mean squared error when PB and B are regarded as a single label.
- Mean Squared Error (NB+PB,B): Mean squared error when NB and PB are regarded as a single label.

chat-dialogue-corpus

[2] https://www.nttdocomo.co.jp/service/developer/smart_phone/analysis/chat/

[3] Fleiss' κ when PB and B are treated as a single label.

[4] http://crowdworks.jp

[5] http://crowdsourcing.yahoo.co.jp

Table 1: Statistics of the datasets.

	Training data		Development/test data	
	init100	**rest1046**	**dev**	**test**
No. of dialogues	100	1046	20	80
No. of user utterances	1,000	10,460	200	800
No. of system utterances	1,100	11506	220	880
No. of words (user)	7,583	78,785	1,998	7,704
No. of words (system)	6,804	69,431	1,665	6,559
No. of annotators	24	2 or 3	30	30
NB (Not a breakdown)	59.2%	58.3%	36.6%	37.1%
PB (Possible breakdown)	22.2%	25.3%	31.3%	32.5%
B (Breakdown)	18.6%	16.4%	32.2%	30.2%
Fleiss' κ (NB, PB, B)	0.28	0.28	0.19	0.20
Fleiss' κ (NB, PB+B)[3]	0.40	0.40	0.27	0.27

These metrics compare the distributions of the labels, thus enabling a direct comparison with the gold labels. However, the results may not be as easily interpretable as the classification-related metrics because they do not directly translate to detection performance.

5. Evaluation workshop

Six teams of participants, hereafter referred to as Team 1 through Team 6, participated in the challenge. Below, we briefly overview the methods of the participants in addition to the baseline we prepared. The names/institutions of the participants are not stated here because the aim of the workshop was not to encourage competition and we hoped anonymization would encourage participation from both academia and industry. Although not linked with the team numbers in this paper, the details of the six teams' methods can be found in (Horii and Araki, 2015; Taniguchi and Kano, 2015; Kobayashi et al., 2015; Mizukami et al., 2015; Sugiyama, 2015; Inaba and Takahashi, 2015).

5.1. Methods

Representing the current trends, four participants used deep learning or deep neural networks (DNNs). There was also one rule-based method and one SVM-based method to round out the six. Our baseline was based on conditional random fields (CRFs) (Lafferty et al., 2001). Although space constraints prevent us from going into detail about the methods, brief descriptions are provided below. In these descriptions, RNN, LSTM, and NCM stand for recurrent neural network, long short-term memory, and neural conversational model (Vinyals and Le, 2015), respectively.

Baseline CRF-based method. The detector labels utterance sequences with the three breakdown labels. The features used are words in the target utterance and its previous utterances. To determine the gold label for training, the baseline uses the same threshold t as in the classification-related metrics.

Team 1 LSTM-RNN-based method. The features used are word vector, co-occurrence frequency vector for words between system and user utterances, and a vector representing word and co-occurrence frequency vectors created by Sent2Vec (an extension of Word2Vec (Mikolov et al., 2013)). Run 1 used RNN and Run 2 used LSTM.

Team 2 LSTM-RNN-based method. The features used are word frequency vectors based on Word2Vec. The runs differ in the structures of the RNN.

Team 3 Rule-based method. Keywords are extracted from user and system utterances and, on the basis of the keywords, heuristic rules are applied to detect whether breakdown has occurred. The runs differ in the rules applied.

Team 4 SVM-based method. The features used are word frequency vectors of the system and previous user utterance. The runs differ in the degree of the kernel used.

Team 5 DNN-based method. The features used are the dialogue act of the system and the previous user utterance, the dialogue act the system should produce next, the perplexity of the system utterance calculated from the word n-grams of valid utterances, and the question classification result for the user utterance. The runs differ in the data to which the parameters were tuned.

Team 6 LSTM-RNN-based method. The features used are the word vector encoded by use of NCM, LSTM, bag-of-word embedding, and an extended NCM. Only one run was submitted.

5.2. Results

Tables 2 and 3 show the results of the submitted runs of the participants in classification-related and distribution-related metrics, respectively.

For the classification-related metrics, Team 5, who adopted a DNN-based method, achieved the highest F-measure. Team 6, who also used deep learning (LSTM-RNN), achieved good precision. Here, Team 5 used extensive external knowledge (dialogue-act annotated corpus, database of questions and answers, etc.), which probably led to its superiority in recall. We should point out that B label is

Table 2: Results of submitted runs of participants when classification-related metrics ($t = 0.5$) were used. Bold font indicates the best performance in each column.

	Accuracy	Precision (B)	Recall (B)	F (B)	Precision (PB+B)	Recall (PB+B)	F (PB+B)
Baseline (t=0.5)	0.481	0.355	0.136	0.197	0.773	0.628	0.693
Team 1 run1	0.530	0.277	0.429	0.337	0.758	0.444	0.560
run2	0.628	0.429	0.015	0.029	0.857	0.011	0.022
Team 2 run1	**0.643**	0.444	0.444	0.444	0.824	0.361	0.502
run2	0.574	0.475	0.237	0.316	0.796	0.468	0.589
run3	0.627	0.443	0.258	0.326	0.836	0.254	0.390
Team 3 run1	0.460	0.167	0.035	0.058	0.725	0.510	0.599
run2	0.515	0.288	0.556	0.379	0.725	0.510	0.599
run3	0.570	0.221	0.126	0.161	0.628	0.131	0.216
Team 4 run1	0.599	0.313	0.232	0.267	0.741	0.201	0.316
run2	0.633	0.414	0.146	0.216	0.800	0.103	0.183
run3	0.627	0.380	0.096	0.153	0.820	0.076	0.138
Team 5 run1	0.515	0.366	0.551	0.440	0.790	0.670	0.725
run2	0.477	0.333	**0.783**	**0.468**	0.760	0.836	0.796
run3	0.442	0.366	0.551	0.440	0.753	**0.849**	**0.798**
Team 6 run1	0.631	**0.531**	0.086	0.148	**0.926**	0.138	0.240

Table 3: Results of submitted runs of participants when distribution-related metrics were used. Bold font indicates the best performance in each column.

	JS divergence			Mean squared error		
	(NB,PB,B)	(NB,PB+B)	(NB+PB,B)	(NB,PB,B)	(NB,PB+B)	(NB+PB,B)
Baseline (t=0.5)	0.399	0.261	0.194	0.212	0.228	0.156
Team 1 run1	0.200	0.122	0.106	0.104	0.133	0.105
run2	0.375	0.357	0.150	0.200	0.366	0.123
Team 2 run1	0.118	0.094	**0.058**	0.069	0.108	**0.058**
run2	0.143	0.106	0.076	0.083	0.118	0.075
run3	0.122	0.098	0.064	0.070	0.109	0.065
Team 3 run1	0.403	0.306	0.178	0.211	0.278	0.143
run2	0.440	0.307	0.313	0.229	0.280	0.289
run3	0.455	0.415	0.231	0.239	0.401	0.197
Team 4 run1	0.429	0.379	0.226	0.224	0.362	0.191
run2	0.420	0.398	0.190	0.218	0.383	0.152
run3	0.423	0.407	0.186	0.220	0.393	0.148
Team 5 run1	0.392	0.248	0.245	0.203	0.213	0.213
run2	0.394	0.212	0.297	0.204	0.172	0.271
run3	0.395	0.212	0.245	0.208	0.173	0.213
Team 6 run1	**0.105**	**0.080**	0.059	**0.060**	**0.090**	0.062

more difficult to classify compared to PB+B; the best F-measure for the former is 0.468 while that for the latter is 0.798. This indicates that currently it is difficult to distinguish between PB and B, but it is possible to distinguish breakdowns from non-breakdowns.

As for the distribution-related metrics, Team 6 achieved the best performance, followed by Team 2. They both utilized LSTM-RNN-based methods. As we discussed in the metrics section, it is difficult to interpret the performance of these detectors from the numeric values. However, it is interesting to note that the best performing team in the classification-related metrics did not perform as well in these metrics. It will be worthwhile to investigate further the relationship between the two types of metric (classification-based and distribution-based). It will also be necessary to examine which type of metric is most suitable in terms of improving end-to-end dialogue systems. For this purpose, we aim to build dialogue systems that utilize these detectors and calculate the correlations between the metrics of these values and the overall performance of the systems.

6. Summary and future work

We described our dialogue breakdown detection challenge in which the task was to detect dialogue breakdowns in chat dialogue. We created datasets, determined the eval-

uation metrics, and held the event. Results of the submitted runs of the participants demonstrate that DNN-based methods work sufficiently well, as they enable breakdowns to be distinguished from non-breakdowns with an F-measure of 0.798. However, it is still difficult to detect severe breakdowns with high accuracy. In future work, we want to pursue the best metrics for dialogue breakdown detection and find relationships between the evaluation metrics we enumerated. We aim to hold a second challenge to further improve the detection performance so that dialogue systems with fewer breakdowns can be achieved. In addition, we want to deal with different languages and modalities, since we only dealt with Japanese and text chat in the challenge reported here.

Bear, J., Dowding, J., and Shriberg, E. (1992). Integrating multiple knowledge sources for detection and correction of repairs in human-computer dialog. In *Proc. ACL*, pages 56–63.

Black, A. W. and Eskenazi, M. (2009). The spoken dialogue challenge. In *Proc. SIGDIAL*, pages 337–340.

Bulyko, I., Kirchhoff, K., Ostendorf, M., and Goldberg, J. (2005). Error-correction detection and response generation in a spoken dialogue system. *Speech Communication*, 45(3):271–288.

Carpenter, P., Jin, C., Wilson, D., Zhang, R., Bohus, D., and Rudnicky, A. I. (2001). Is this conversation on track? In *Proc. Eurospeech*, pages 2121–2124.

Higashinaka, R., Funakoshi, K., Araki, M., Tsukahara, H., Kobayashi, Y., and Mizukami, M. (2015a). Towards taxonomy of errors in chat-oriented dialogue systems. In *Proc. SIGDIAL*, pages 87–95.

Higashinaka, R., Funakoshi, K., Mizukami, M., Tsukahara, H., Kobayashi, Y., and Araki, M. (2015b). Analyzing dialogue breakdowns in chat-oriented dialogue systems. In *Proc. ERRARE*.

Higashinaka, R., Mizukami, M., Funakoshi, K., Araki, M., Tsukahara, H., and Kobayashi, Y. (2015c). Fatal or not? Finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems. In *Proc. EMNLP*, pages 2243–2248.

Horii, T. and Araki, M. (2015). A breakdown detection method based on taxonomy of errors in chat-oriented dialogue. In *JSAI Technical Report (SIG-SLUD-75-B502)*, pages 33–36. (in Japanese).

Inaba, M. and Takahashi, K. (2015). Dialogue breakdown detection using long short-term memory recurrent neural network. In *JSAI Technical Report (SIG-SLUD-75-B502)*, pages 57–60. (in Japanese).

Kobayashi, S., Unno, Y., and Fukuda, M. (2015). Multitask learning of recurrent neural network for detecting breakdowns of dialog and language modeling. In *JSAI Technical Report (SIG-SLUD-75-B502)*, pages 41–46. (in Japanese).

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. ICML*.

Martinovsky, B. and Traum, D. (2003). The error is the clue: Breakdown in human-machine interaction. In *Proc. ISCA Workshop on Error Handling in Spoken Dialogue Systems*, pages 11–16.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*, pages 3111–3119.

Mizukami, M., Sugiyama, K., Neubig, G., Yoshino, K., Sakti, S., and Nakamura, S. (2015). Construction of rnnbsed dialogue breakdown detector. In *JSAI Technical Report (SIG-SLUD-75-B502)*, pages 47–50. (in Japanese).

Onishi, K. and Yoshimura, T. (2014). Casual conversation technology achieving natural dialog with computers. *NTT DOCOMO Technical Jouranl*, 15(4):16–21.

Sugiyama, H. (2015). Chat-oriented dialogue breakdown detection based on combination of various data. In *JSAI Technical Report (SIG-SLUD-75-B502)*, pages 51–56. (in Japanese).

Taniguchi, R. and Kano, Y. (2015). Construction of automatic detector for dialogue breakdowns based on rules with keywords extraction. In *JSAI Technical Report (SIG-SLUD-75-B502)*, pages 37–40. (in Japanese).

Vinyals, O. and Le, Q. (2015). A neural conversational model. In *Proc. ICML Deep Learning Workshop*.

Williams, J., Raux, A., Ramachandran, D., and Black, A. (2013). The dialog state tracking challenge. In *Proc. SIGDIAL*, pages 404–413.