Dialogue Breakdown Detection based on Estimating Appropriateness of Topic Transition

Hiroaki Sugiyama[1]

[1]NTT Communication Science Laboratories, Japan

sugiyama.hiroaki@lab.ntt.co.jp

Abstract

Chat-oriented dialogue systems sometimes generate inappropriate response utterances to user utterances that cause dialogue breakdown. If we can detect such inappropriate utterances and suppress them, it helps to continue the dialogue. Although the estimation of the appropriateness of topic transition is an important factor for the breakdown detection, previous state-of-the-art dialogue breakdown detector leverages only the topic similarities between utterance pairs, which is not enough to evaluate natural jump of topics. In this paper, we propose novel features to assess the topic transition, and examine the effectiveness of the features for improving the performance of dialogue breakdown detection.

Index Terms: dialogue breakdown detection, chat-oriented dialogue system, appropriate topic transition

1. Introduction

Chatting with people is an important function of dialogue systems in building social relationships with users. This not only provides therapeutic and entertainment benefits but also plays an important role in drawing out the user's potential requirements and constructing a good relationship with the user. Furthermore, such conversational dialogue has the potential to improve the performance of task-oriented dialogue [1]. Thus, the construction of conversational dialogue systems (also called non-task oriented dialogue systems or chat-oriented dialogue systems) has recently gained attention [2, 3, 4].

Chat-oriented dialogue systems need to respond to a very wide range of topics expressed by user utterances. It is difficult for the current dialogue systems to continue outputting appropriate responses, and thus utterances that cause the dialogue to collapse are often generated. It is assumed that continuation of dialogue becomes easy when we can detect and suppress such problematic utterances.

In a previous dialogue breakdown detection challenge (DBDC), the author proposed a dialogue breakdown detection system that captures frequently appearing error patterns that are specific to each utterance generation approach [5]. For example, in a generation-based approach of fitting words to some template, it is easy to generate system utterances that have topics related to the user utterance. However, such approach sometimes generates problematic utterances when the templates are applied to handle unexpected words [6, 7]. In the retrieval-based approach, although non-sentences are not easily generated, system utterances that do not match the topics are likely to be generated when the system cannot search for utterances that suitably match the user utterances. The author's previous examinations found that these errors can be captured with word match rates and word2vec-based distances such as the distances of averaged word vectors and the word mover's distances [8]) between user and system utterances.

However, such a dialogue breakdown detection system has the following limitation. The system cannot evaluate the appropriateness of topic transitions because the system utilizes only the exact word match or distances obtained by word2vec. For example, if a user utterance is *Would you like to go see a movie?* and a system utterance is *I'd like to eat popcorn!*, the system evaluates the system utterance as inappropriate, whereas we think the transition from *movie* to *popcorn* is natural. In this research, we examine several features that are useful for evaluating such topic transitions with the previous DBDC dataset [9].

2. Proposed features and algorithms

In this section, we explain our previously proposed method as well as newly proposed features and learning algorithms. The data distributed in the DBDC to be analyzed in this research contains text chats between users and one of the following three systems, and evaluation annotations (NB: not a breakdown, PB: possibly breakdown, B: breakdown) annotated by 30 persons.

DCM Conversation system API produced by NTT docomo.

DIT Conversation system that utilizes label propagation for topic transitions produced by DENSO IT laboratories [7, 10].

IRS Example-based conversation system (almost the same as IR-status [2]) provided by the DBDC organizer.

From observing the output examples, DCM is a generation-based approach that fits predicate-argument pairs related to a user utterance to generic utterance templates. IRS is a retrieval-based approach that outputs human-generated examples, while DIT is a hybrid approach of example- and template-based approaches that retrieve utterances related to the user utterances and substitute unrelated parts of the sentences with more suitable words.

2.1. Proposed features

Table 1 shows the proposed features and their descriptions. We categorize the features into the following feature groups.

Word-based similarities In DIT and IRS, there are cases where utterances with topics completely different from user utterances are generated. In DCM, the system sticks to a specific topic, and, as a result, there are many cases where an utterance with almost the same contents repeatedly occurs. In order to detect these errors, we define word-based similarity features between system and user utterances and between the target and a previous system utterance. The word similarities are calculated with three types of methods: the cosine similarity of the bag-of-word vectors, the word mover's distance (wmd), and the cosine similarity of averaged word vectors. In

Table 1: *Proposed features (New means that the feature is newly introduced in this DBDC3 and not used in our previous method).*

Feature group	Feature name	New	Description
Word-based similarities	matchrate		Content word matching rate
	wmd		Word mover's distance between target and previous two utterances (all).
	wmd-noun	✓	Word mover's distance between target and previous two utterances (noun).
	wmd-nv	✓	Word mover's distance between target and previous two utterances (noun and verb).
	wmd-prev	✓	Word mover's distance between previous two utterances (calculated with four word types: all, noun, verb, noun and verb).
	wmd-comb	✓	Word mover's distance between target and concatenated previous two utterances.
	w2vd		Cosine similarities of averaged word vectors obtained by word2vec[1] between target and user utterances.
Dialogue act	DA		Dialogue acts to be estimated for target and user utterances, and predicted to be suitable for the following utterance of user utterance.
	ADA	✓	Abstracted dialogue acts from original 33 tag types into eight.
Sentence length	toks-length		Number of words.
	sent-length		Number of characters.
Number of elapsed turns	n-turns		Number of elapsed turns.
Language model	LM-ug	✓	Normalized logprobs of language models of target utterance with uni-gram probabilities. Models are trained with Twitter corpus and Wikipedia corpus.
Sentence embedding	s2s	✓	Seq2seq encode vector of target utterance (dim: 500).
	s2s-1	✓	Seq2seq encode vector of user utterance (dim: 500).
	s2s-2	✓	Seq2seq encode vector of previous system utterance (dim: 500).
Interrogative	interrogative	✓	Type of interrogative of user utterance (when, which, who, where, why, what, how and None)
PDB response	pdb-res	✓	Word distance between target utterance and questions similar to user utterance retrieved from Person DB [11].
IDF	prev-idf	✓	Position of utterances that contain a high-IDF word appearing in a user utterance.
	large-idf	✓	Existence of a new high-IDF word in target utterance.
Abstracted content words	abst-words	✓	Bag of abstracted content words.

this DBDC3, to focus on the transition of topics, we add wmd-based features that are calculated only with content words such as noun and verb without stopwords (wmd-noun, wmd-nv and wmd-comb). We also add wmd-prev, which is word mover's distance calculated between previous two utterances with four word types: all, noun, verb, noun and verb. This captures whether the dialogue topic has just changed or not.

Dialogue act All of the systems sometimes respond with questions even when the user utterance is a question. Besides, it seems that the systems lack the function to answer questions. In order to detect these errors, we utilize estimated dialogue acts of the target system utterance and the user utterance. In addition, we use dialogue acts that are expected to be suitable for the next utterance after the user utterance. In this research, we use a dialogue acts definition proposed in [12], in which they categorize utterances into 33 dialogue acts. The dialogue acts estimator was learned with NTT's chat dialogue corpus (3680 dialogues) [4], using linear SVM and word 1,2-gram features [13]. In DBDC3, we add abstracted dialogue acts (ADA) that shrink from original 33 tag types into eight to alleviate data sparseness.

Sentence length With the technology of the current dialogue system, it is difficult to estimate the consistency of the user utterance and the system utterance. Therefore, there is the problem that the longer the system utterance is, the greater the possibility that an unrelated element is included. In particular, DIT tends to generate very long utterances, and thus, the utterance does not match the content of the user utterance. Here, we add word length and character length of the target utterance to the features.

Number of elapsed turns All three dialogue systems generate relatively appropriate utterances at the beginning of a dialogue, but the proportion of inappropriate utterances tends to increase as the dialogue proceeds. Therefore, the number of elapsed turns from the start of the dialogue is added to the features.

Language model In DCM and DIT, sentences that contain unnatural collocations are frequently generated and annotated as breakdown. In order to evaluate the fluency of the utterances, we newly adopt the log probability of the utterances based on language models, whose values are normalized (divided) with each word probability.

Sentence embedding In DBDC2, some teams showed the effectiveness of seq2seq for the dialogue breakdown detection task [9]. In DBDC3, we introduce encoded vectors of the utterances based on a seq2seq model, which is trained with the NTT's chat dialogue corpus.

Interrogative When a user utterance contains an interrogative, the system utterance should adhere to the appropriate way of answering the specific interrogative. Since our previous method did not take into account the differences in how to answer questions, we newly introduce the interrogative types in the features.

PDB response Natural topic transition is difficult to capture with only the word-embedding-based distances between target and user utterances. We leverage many question-answer pairs about the talker's personality, called Person DB (PDB) [11]. The highest IDF words of a user utterance are extracted to retrieve questions that contain the word. In this DBDC3, we calculate the word distances

Table 2: *Differences of evaluation scores when each feature is excluded (raw value / difference). Bold items are the five-best features, and underlined items are those that deteriorate performance.*

Feature group	Excluded feature	Accuracy	Mean squared error	JS divergence
	None	0.58764	0.03861	0.07285
Word-based similarities	matchrate	0.58570 / -0.00194	0.03881 / +0.00020	0.07323 / +0.00038
	wmd	0.58994 / +0.00230	0.03876 / +0.00015	0.07312 / +0.00027
	wmd-noun	**0.58400 / -0.00364**	0.03890 / +0.00029	0.07330 / +0.00045
	wmd-nv	**0.58412 / -0.00352**	0.03870 / +0.00008	0.07297 / +0.00013
	wmd-prev	0.58570 / -0.00194	0.03882 / +0.00021	0.07320 / +0.00036
	wmd-comb	0.58594 / -0.00170	0.03871 / +0.00010	0.07303 / +0.00018
	w2vd	0.58642 / -0.00121	0.03865 / +0.00004	0.07292 / +0.00008
Dialogue act	DA	0.58982 / +0.00218	0.03874 / +0.00013	0.07310 / +0.00025
	ADA	0.59115 / +0.00352	0.03890 / +0.00028	0.07332 / +0.00047
Sentence length	toks-length	0.58812 / +0.00048	0.03864 / +0.00003	0.07290 / +0.00005
	sent-length	0.58691 / -0.00073	0.03871 / +0.00010	0.07301 / +0.00017
Number of elapsed turns	n-turns	**0.58424 / -0.00339**	0.03878 / +0.00017	0.07314 / +0.00029
Language model	LM-ug	**0.58121 / -0.00642**	**0.03957 / +0.00096**	**0.07450 / +0.00165**
Sentence embedding	s2s	0.58812 / +0.00048	**0.03957 / +0.00096**	**0.07449 / +0.00164**
	s2s-1	0.59200 / +0.00436	**0.03916 / +0.00055**	**0.07376 / +0.00092**
	s2s-2	0.59042 / +0.00279	**0.03900 / +0.00039**	**0.07354 / +0.00069**
Interrogative	interrogative	0.58667 / -0.00097	0.03862 / +0.00001	0.07291 / +0.00006
PDB response	pdb-res	0.59152 / +0.00388	**0.03901 / +0.00040**	**0.07356 / +0.00071**
IDF	prev-idf	0.58788 / +0.00024	0.03867 / +0.00006	0.07301 / +0.00016
	large-idf	**0.58436 / -0.00327**	0.03873 / +0.00012	0.07303 / +0.00019
Abstracted content words	abst-words	0.58618 / -0.00145	0.03879 / +0.00018	0.07310 / +0.00025

Table 3: *Differences of evaluation scores when each feature group is excluded. Bold items are the three-best features, and underlined items are those that deteriorate performance.*

Excluded feature group	Excluded features	Accuracy	Mean squared error	JS divergence
-	None	0.58764	0.03861	0.07285
Word-based similarities	matchrate,w2vd,wmd,wmd-comb,wmd-noun,wmd-nv,wmd-prev	0.58655 / -0.00109	**0.03960 / +0.00098**	**0.07454 / +0.00170**
Dialogue act	ADA,DA	0.58873 / +0.00109	**0.03947 / +0.00085**	**0.07429 / +0.00144**
Sentence length	sent-length,toks-length	**0.58618 / -0.00145**	0.03880 / +0.00019	0.07318 / +0.00033
Number of elapsed turns	n-turns	**0.58424 / -0.00339**	0.03878 / +0.00017	0.07314 / +0.00029
Language model	LM-ug	**0.58121 / -0.00642**	**0.03957 / +0.00096**	**0.07450 / +0.00165**
Sentence embedding	s2s,s2s-1,s2s-2	0.59103 / +0.00339	0.03922 / +0.00061	0.07421 / +0.00137
Interrogative	interrogative	0.58667 / -0.00097	0.03862 / +0.00001	0.07291 / +0.00006
PDB response	pdb-res	0.59152 / +0.00388	0.03901 / +0.00040	0.07356 / +0.00071
IDF	large-idf,prev-idf	0.58885 / +0.00121	0.03868 / +0.00007	0.07302 / +0.00017
Abstracted content words	abst-words	**0.58618 / -0.00145**	0.03879 / +0.00018	0.07310 / +0.00025

between the words appearing in the extracted questions and those appearing in the target utterance and then add the minimum distance among them to the features.

IDF When a new high-IDF word appears in the target utterance, this indicates the possibility that the dialogue topic has changed. In addition, if a high-IDF word appearing in the user utterance also appears in previous and target utterances, the system may stick to the word as the dialogue topic. We introduce such word appearances (higher IDF than threshold) to the features. The IDF values are calculated from our year-long Twitter corpus.

Abstracted content words With a small number of training data, it is difficult to model the topic transitions, especially when the dialogue topics widely vary. Therefore, we propose a new method that captures the topic transition patterns. First, we convert the content words (noun, verb and adjective) in each utterance with an ID based on POS, such as Noun 1, Verb 1, in the order of their appearance. Second, we convert each utterance into a vector where the corresponding bit to a content word is 1. Finally, we concatenate the converted utterance vectors into a single vector.

2.2. Training data

In this task, the distribution-related metrics such as JS-divergence or mean squared errors are emphasized. We expect that matching distribution is more sensitive to the characteristics of the data than simple F-value evaluation. In fact, in the preliminary experiment where we estimate the distribution of DBDC2-eval, the best performance is obtained when the model is learned with only the DBDC1-dev, eval and DBDC2-dev. When we added rest1046 (only two persons evaluated) and init100 (researchers evaluated instead of ordinal persons such as in DBDC2) distributed in DBDC1, the performance greatly deteriorated. Therefore, we do not use additional learning data in this challenge, but use only distributed DBDC1-dev, eval and DBDC2-dev for model training.

Table 4: *Comparison of ensemble regression models trained with all features*

Model	Accuracy	Mean squared error	JS divergence
a) 1: [PASS], 2:ETR	0.58679	0.03908	0.07436
b) 1: [PASS, KNN-dist], 2: ETR	0.58461	0.03913	0.07444
c) 1: [PASS, KNN-dist, t-SNE], 2: ETR	0.58776	0.03911	0.07443
d) 1: [PASS, KNN-dist, RFR, ETR, KNR, GBR, SVR], 2: ETR	0.58764	**0.03861**	**0.07285**
e) 1: [PASS, KNN-dist, t-SNE, RFR, ETR, KNR, GBR, SVR], 2: ETR	**0.59055**	0.03868	0.07298

2.3. Algorithm

In our previous method, we utilized ExtraTreesRegressor to effectively leverage the combination of features for the estimation of evaluation distributions. Although this method can more effectively estimate the distributions than DNN-based methods with small-scale data, other regression methods such as Support Vector Regressors (SVR) or k-nearest neighbor regressor (KNR) also have the potential to estimate them. To leverage the output of such methods, we adopt a stacked regression model [14] that utilizes predictions of base regressors as the input features of higher-level regressors. This stacking approach is widely used in competitions such as kaggle. We examined the effectiveness of stacked regression by several regression methods: Random forest regressor (RFR), Extra trees regressor (ETR), K-nearest Neighbor regressor (KNR), Gradient boosting regressor (GBR), Support vector regressor (SVR), and naive minimum distances with samples for each annotated label (KNN-dist). In addition, we introduced two types of feature passing models: a PASS model that copies the input feature as it is and t-SNE [15] for dimension reduction. We used ETR as the top-layer regressor, since other simple models such as linear regressions did not work well in a preliminary experiment. Here, we utilized scikit-learn[2] for the implementation of these regression models.

3. Experiment

3.1. Experiment settings

We combine the features described in the previous chapter to construct dialogue breakdown detectors and compared their breakdown detection performances. Since importance is placed on distribution-related metrics (Mean Squared Error and JS-divergence), we examined the features that minimize distribution distances. In this research, we examined the effectiveness of the features by subtracting certain features from the case of using all features. We used DBDC1-dev, DBDC1-eval and DBDC2-dev for the training data and DBDC2-eval for evaluation data.

3.2. Results

Table 2 shows the results of the experiment. Each row shows an excluded feature's name, the evaluation scores and their differences from the base scores calculated with all of the features. If the value is worse than the base scores (lower accuracy or higher distribution-related metrics), the feature plays an important role in estimation. Conversely, as the value is better (higher accuracy or lower distribution-related metrics), the feature deteriorates performance, Bold items are the five best features for each metric, and the underlined items are those that deteriorate performance.

Table 2 shows that the base scores with all of the features provide the best performance in distribution-related metrics (mean squared error and JS divergence) as well as top-5 performance in accuracy. This also illustrates that, LM-ug (normalized logprob of LM) and wmd-noun are effective for improving accuracy. For the distribution-related metrics, LM-ug, s2s/s2s-1 (sentence embedding features), pdb-res (natural word transition patterns) are important for improving estimation performance.

Table 3 shows the estimation performance when each feature group is excluded. It indicates that dialogue act and word-based similarity feature groups are effective for improving the performance.

Table 4 illustrates the difference among the ensemble settings of the estimators. *Model* column shows the structure of each model (the numbers mean their layer numbers). Table 4 shows that all models are not so different in accuracy. On the other hand, models that have complicated structures such as d) and e) achieved the best scores in the distribution-related metrics.

[2]http://scikit-learn.org/

Table 5: *Evaluation scores for DBDC3-eval (overall)*

Runs	Acc	MSE	JSD	F1 (B)	F1 (PB+B)
Run 1	**0.6129**	**0.0371**	**0.0691**	**0.6714**	0.7918
Run 2	0.6085	0.0373	0.0693	0.6684	**07909**
Run 3	0.6017	0.0388	0.0719	0.6641	**0.8055**

Table 6: *Evaluation scores for DBDC3-eval (DIT data only)*

Runs	Acc	MSE	JSD	F1 (B)	F1 (PB+B)
Run 1	0.6181	0.0283	0.0524	0.7306	0.8341
Run 2	0.6127	0.0285	0.0528	0.7307	0.8335
Run 3	**0.6236**	**0.0279**	**0.0516**	**0.7306**	**0.8410**

3.3. Settings and results of submitted data for DBDC3

The data submitted to the challenge task DBDC3 were developed in the following three settings, which show high performance of the distribution-related metrics.

Run 1 Model: d), features: all features, training data: DBDC1-dev, DBDC1-eval, DBDC2-dev and DBDC2-eval, test data: DBDC3-eval.

Run 2 Model: e), features: all features, training data: DBDC1-dev, DBDC1-eval, DBDC2-dev and DBDC2-eval, test data: DBDC3-eval.

Run 3 Model: d), features: all features, training data: DBDC2-dev and DBDC2-eval, test data: DBDC3-eval.

Table 5 illustrates that run 1 shows the highest performance for all metrics except for F1 (PB+B). The model cannot show high F1 (PB+B), in comparison with other teams' systems [16], because our models tend to output NB when NB's and PB's probabilities have close values.

In Table 6, run 3 shows the highest scores in DIT, contrary to the overall results where run 3 shows the lowest score among our three runs. This is because run 3 uses DBDC2 data, where the numbers of DCM, DIT and IRS data are the same (DBDC1-dev and eval contain only DCM data).

4. Conclusion

In this paper, we analyzed the features and the ensemble of regression models useful for detecting the breakdown of chat dialogue. In particular, by introducing features that capture the naturalness of topic transitions, we achieved dialogue breakdown detection with higher accuracy and more similar distributions than achieved by conventional methods. In addition, we showed that seq2seq-based embedded vectors of utterances learned with large scale dialogue data and word pairs frequently appearing in question-response patterns were useful for improving distribution-related metrics.

As future work, it is necessary to clarify which kinds of topic transitions can be handled correctly. By analyzing actual cases in detail, we will examine which kinds of transitions can be captured to achieve a more stable dialogue breakdown detector.

5. References

[1] T. Bickmore and J. Cassell, "Relational Agents: A Model and Implementation of Building User Trust," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2001, pp. 396–403.

[2] A. Ritter, C. Cherry, and W. B. Dolan, "Data-Driven Response Generation in Social Media," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 583–593.

[3] W. Wong, L. Cavedon, J. Thangarajah, and L. Padgham, "Strategies for Mixed-Initiative Conversation Management using Question-Answer Pairs," in *Proceedings of the 24th International Conference on Computational Linguistics*, 2012, pp. 2821–2834.

[4] R. Higashinaka, K. Imamura, T. Meguro, C. Miyazaki, N. Kobayashi, H. Sugiyama, T. Hirano, T. Makino, and Y. Matsuo, "Towards an open-domain conversational system fully based on natural language processing," in *Proceedings of the 25th International Conference on Computational Linguistics*, 2014, pp. 928–939.

[5] Hiroaki Sugiyama, "Chat-oriented Dialogue Breakdown Detection based on the Analysis of Error Patterns in Utterance Generation," in *Proceedings of SIG-SLUD*, 2016, pp. 81–84 (in Japanese).

[6] H. Sugiyama, T. Meguro, R. Higashinaka, and Y. Minami, "Open-domain Utterance Generation using Phrase Pairs based on Dependency Relations," in *Proceedings of the 2014 IEEE Workshop on Spoken Language Technology*, 2014, pp. 60–65.

[7] H. Tsukahara and K. Uchiumi, "System utterance generation by label propagation over association graph of words and utterance patterns for open-domain dialogue systems," in *Proceedings of Pacific Asia Conference on Language, Information and Computation*, 2015.

[8] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger, "From Word Embeddings To Document Distances," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015, pp. 957–966.

[9] Ryuichiro Higashinaka, Kotaro Funakoshi, Michimasa Inaba, Yuki Arase and Yuiko Tsunomori, "Dialogue Breakdown Detection Challenge 2," in *Proceedings of SIG-SLUD*, 2016 (in Japanese).

[10] Hiroshi Tsukahara and Kei Uchiumi, "Improving Chat Generation Method by Label Propagation Using Dialogue Act and Topic Estimation," in *Proceedings of the 30th Annual Conference of the Japanese Society for Artificial Intelligence*, 2016 (in Japanese).

[11] H. Sugiyama, T. Meguro, and R. Higashinaka, "Large-scale Collection and Analysis of Personal Question-answer Pairs for Conversational Agents," in *Proceedings of International Conference on Intelligent Virtual Agents*, 2014, pp. 420–433.

[12] H. Sugiyama, T. Meguro, R. Higashinaka, and Y. Minami, "Open-domain Utterance Generation for Conversational Dialogue Systems using Web-scale Dependency Structures," in *Proceedings of the 14th annual SIGdial Meeting on Discourse and Dialogue*, 2013, pp. 334–338.

[13] L. Breiman, "Stacked regressions," *Machine learning*, vol. 24, no. 1, pp. 49–64, 1996.

[14] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[15] R. Higashinaka, K. Funakoshi, M. Inaba, Y. Tsunomori, T. Takahashi, and N. Kaji, "Overview of Dialogue Breakdown Detection Challenge 3," in *Proceedings of Dialogue System Technology Challenge 6 (DSTC6) Workshop*, 2017.