End-to-end Character-Level Dialogue Breakdown Detection with External Memory Models

Taichi Iki[1] *, *Atsushi Saito*[1] *

[1]Nextremer Co., Ltd., Tokyo, Japan

taichi.iki@nextremer.com, atsushi.saito@nextremer.com

Abstract

Real human-machine dialogues can inevitably end up with breakdown because no rule covers all situations. In particular, performance of the dialogue breakdown detection task for open domain conversation encounters the following technical difficulties: 1) dialogue histories often contain words which don't appear directly in training data, and 2) preparing and annotating a large amount of conversation data require human resources. Here, we propose end-to-end memory-network-based detection models having character-level sentence embedding to handle unseen words and an external memory mechanism to integrate information in the local dialogue context. Moreover, to tackle issues of rareness of annotated data, we employ an unsupervised method that is a task to predict the next sentence in an unannotated document. The results of experiments with a Japanese-language data set indicate that integrating local-context memory information contributes to accuracy improvement.

Index Terms: dialogue breakdown detection, end-to-end neural detection model

1. Introduction

Recently, the need for chat-oriented natural conversational interfaces has been increasing. As a result, dialogue breakdown detection, of which Martinovsky et al. provide a definition [1], is becoming more important. Therefore, Higashinaka et al. have initiated construction of a unified format of annotated chat-oriented data sets for dialogue breakdown detection [2][3].

In DBDC3, dialogue breakdown detection is formulated as a three-class classification problem of system utterances. The class labels consist of not a breakdown (NB), possible breakdown (PB) and breakdown (B). A detector predicts the class label of given system utterance with a distribution over class labels. Dialogue history up to the target utterance is also inputted as a series of utterances.

Recent work on dialogue systems with neural networks includes a promissing approach, end-to-end memory networks (MemN2N) [4]. MemN2N models were originally developed in challenges to construct models to perform tasks in the area of question answering, along with models that can handle long-term dependencies in sequential data. These kinds of models are naturally applicable to several dialogue tasks. For example, Bordes et al. [8] have proposed methods applicable to goal-oriented dialogue tasks and Lowe et al. [9] have used the term "next utterance classification" to describe settings that allow models to select the next turn from candidate sentences. However, currently, no MemN2N neural model as a strcuture of dialogue breakdown detecter has not been reported.

The settings of tranining tasks is also important in addition to model structures. Some multi-task approaches for breakdown detection have already been proposed. Language model is jointly optimized in [6] and utterence generation is combined with breakdown detection in [7].

In this paper, we propose several models based on MemN2N to perform dialogue breakdown detection, and propose training methods using non-annotated data.

The differences between the models proposed in this study and original MemN2N are the following: 1) character-level sentence embedding, 2) use of *attention over attention* [5] and convolutional networks for the attention module, and 3) use of convolutional networks for reducing memories to fixed-size vectors. The overall architecture of our proposed models is shown in Figure 1.

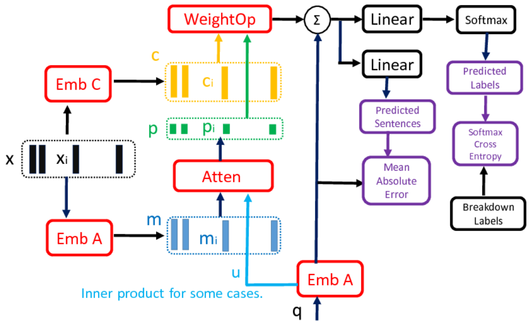

Figure 1: *Architecture of proposed models. Embedding, attention, and weighting operation for memory reduction are abbreviated as* Emb, Atten, *and* WegihtOP, *respectively.*

Character-level embedding can extract word independent features of an utterance, even if the history contains unseen words. Convolutional networks for attention and memory reduction integrate consecutive utterances in a dialogue history as local contexts. Using convolutional networks with small kernel size is based on the hypothesis that, for conversations with no specific goal, single-turn comfort depends on the several preceding turns.

We also use unsupervised training to predict the next sentence for a given sentence in a single passage sampled from wikipedia. Our reasoning for adopting such approaches can be intuitively explained as follows. All dialogue histories have two kinds of sequential information: dependencies associated with turns and those associated with each utterance. Not only dialogue data but also other natural language resources such as passages sampled from wikipedia have the former dependencies. Thus, prediction of a valid sequence of sentences is expected to be a complementary task to breakdown detection and to prevent breakdown detector from overfitting caused by the small amount of dialogue data. Note that several methods to train language modeling or neural conversation in addition to simultaneous breakdown detection training have been proposed in [6] and [7].

* The authors are listed in alphabetical order.

2. Approach

2.1. End-to-end Memory Networks for Dialogue Tasks

A MemN2N model takes (1) a set of sentences $x_1, ..., x_n$ as contents of the external memory and (2) another sentence q as a query, where each sentence is a sequence of tokens (words or characters), before (3) outputting a final predition a.

In a dialogue case, the last utterance given by users or systems are regarded as the query q, and the other past interactions are converted into sentence embeddings to be stored in the external memories. We assume that an input set $x_1, ..., x_i$ is converted into the set of sentence embedding vectors $m_1, ..., m_i$ of dimension d.

In the simplest case, we can take an embedding as an embedding matrix A (of size $d \times V$; where V denotes the number of tokens in vocabulary). These ms are used to compute attention over external memory. The contents of xs are represented via another sentence embedding via an embedding matrix C. Here, q is also embedded (again, via another embedding matrix B with the same shape as A) to obtain an internal query representation u. The overall calculation to obtain the fixed-size vector o from the external memory is as follows:

$$\begin{aligned} p_i &= \mathrm{Softmax}(u^T m_i), && (2.1) \\ o &= \sum_i p_i c_i. && (2.2) \end{aligned}$$

The final prediction $\hat{a}$ is computed for o and u as follows:

$$\hat{a} = \mathrm{Softmax}(W(o + u)), \qquad (2.3)$$

, where W is a parameter matrix to reshape the sum of o and u so that each row of $\hat{a}$ corresponds to a class label.

Note that, although the original MemN2N can input the vector $u_1 = u + o$ to the same structure having multiple memory layers, which are called "hops" (see [4] for more details), in this paper, we present a single-hop model only.

2.2. Preliminaries for Proposed Models

We give a generalized description of the components of the MemN2N structure to describe the proposed models in detail.

As discussed below, a model includes three sentence embeddings. Let the embeddings be EmbA, EmbB and EmbC. EmbA and EmbC map a sentence in dialogue histories to a vector for *memory attention* and *memory reduction* respectively. EmbB converts query sentences to embedded vectors. We note that the three embeddings have the same output shape because of inner product in (2.1) and addition in (2.3) These maps are linear (i.e., matrices) in the original MemN2N; however, in the proposed models, we replace these linear maps with neural networks.

Moreover, we extend inner product in (2.1) and weighted sum in (2.2) to neural networks: Atten maps a pair of an internal query representation and a set of sentence embeddings for memory attention to a distribution of attention over the external memory while WeightOp calculate a fixed-size vector with a given pair of the distribution and a set of sentence embeddings for memory reduction.

Let $a = (a_1, ..., a_d)$ be a real vector of dimension d and f be a map, the domain of which contains a. We denote $\hat{f}(a) = (f(a_1), f(a_2), ..., f(a_d))$. Therefore, we can present the following generalized description of our proposed models:

$$\begin{aligned} m_i &= \widehat{\mathrm{EmbA}}(x_i), && (2.4) \\ u &= \widehat{\mathrm{EmbB}}(q), && (2.5) \\ c_i &= \widehat{\mathrm{EmbC}}(x_i), && (2.6) \\ p &= \mathrm{Atten}(u, m), && (2.7) \\ o &= \mathrm{WeightOp}(p, c), && (2.8) \\ \hat{a} &= \mathrm{softmax}(W(o + u)). && (2.9) \end{aligned}$$

Note that parameter W for the final prediction can be extended to neural networks. We also that the softmax function in (2.9) is not applied if the outputs are predicted sentence vectors instead of distributions over class labels(See Figure 1).

3. Details of Proposed Models

3.1. Sentence Embedding Using Convolutional Networks

Three embeddings are convolutional neural networks having three layers, and are described in Table 1.

Table 1: *Networks computing* EmbA, EmbB, EmbC

layer	# of in-ch	# of out-ch	kernel	stride
1st-3	100	100	(3,1)	1
1st-5	100	100	(5,1)	1
2nd	100	100	(5,1)	1
3rd	100	100	(5,1)	1

An arbitrary (k,1)-kernel takes adjacent k-characters in a given sentence as an input as shown in Figure 2. The structures in Figs. 2(a) and (b) were used for experiments involving convolutional attention (Section 4.4) and for Dialogue Breakdown Detection Challenge 3 (DBDC3) formal runs (Section 4.3), respectively.

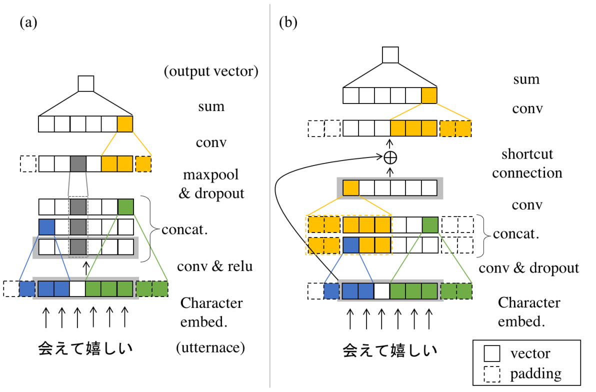

Figure 2: *Two kinds of convolutional embeddings*

We note that, strictly speaking, embeddings are concatenated to an additional index of their speaker: 0 and 1 are assigned to systems and users, respectively. Thus, the number of input channels increases from 100 to 101.

3.2. Memory Attention

3.2.1. Attention Over Attention

The concept of *attention over attention* (AoA) was proposed by Cui et al. [5]. There are two motivations for the AoA mechanism: (1) Both *query-to-document* attention and *document-to-query* attention are necessary to solve reading comprehension tasks without complex hyper-parameter tuning; and (2) these two attention types make it possible to exploit mutual information between the document and query.

We define m as $[m_1, ..., m_i, ..., m_{\widehat{m}}]$, where m_i is sentence embedding in (2.4) and the shape of m is $d \times \widehat{m}$. We can obtain another matrix of the same shape as m by applying the softmax to each row (column) of m , and call the operation $\mathrm{Softmax}_{\mathrm{row}}(m)$ ($\mathrm{Softmax}_{\mathrm{col}}(m)$).

Let R and C be $\mathrm{Softmax}_{\mathrm{row}}(m)$ and $\mathrm{Softmax}_{\mathrm{col}}(m)$, respectively. Thus, the AoA function is calculated as follows:

$$\bar{C} = \frac{1}{\widehat{m}} \sum_t C_{\cdot,t}, \tag{3.1}$$

$$p = \bar{C}^\top \cdot R, \tag{3.2}$$

where $C_{\cdot,t}$ is the t-th column vector of C and $\cdot$ is the matrix product operator. We note that p in (3.2) is an output corresponding to the output of Atten in (2.7).

3.2.2. *Convolutional Attention*

A convolutional attention function is an attention function implemented through convolutional neural networks, and integrates information of several consecutive turns into representations of local context.

Let $\mathrm{broadcast}(m, u)$ be a function calculating a tensor for given m and u, such that calculated tensor and m have compatible shapes with repeating u in the tensor.

The function calculating $\mathrm{Atten}(u, m)$ is obtained as follows. For given u and m,

$$m' = \mathrm{broadcast}(m, u) \odot m, \tag{3.3}$$

$$\tilde{p} = W_{att} \cdot \mathrm{ConvAtten}(m') + b_{att}, \tag{3.4}$$

$$p = \mathrm{Softmax}(\tilde{p}), \tag{3.5}$$

where $\odot$ is the element wise multiplication operator and $\mathrm{ConvAtten}$ in (3.4) is a function computed by a convolutional neural network, the structure of which is described in Table 2.

3.3. Memory Reduction

As a choice of $\mathrm{WeightOp}$ component in (2.8), we can take the following convolutional operation:

$$o = W_{ws} \cdot \mathrm{ConvWeightedSum}(c) + b_{ws}, \tag{3.6}$$

where we define ConvWeightedSum in (3.6) as a function computed by the convolutional neural network whose structure is described in Table 2.

Table 2: *Convolutional network parameters*

layer	# of in-ch	# of out-ch	kernel	stride
		ConvAtten		
1st	1	25	(101,3)	1
		ConvWeightedSum		
1st	1	25	(101,2)	1

The number of output channels corresponds to the memory size. For an arbitrary (101, k)-kernel, the adjacent k-turns in the given dialogue history are taken.

3.4. Final Predictions

We employed a linear layer for the final predictions. The proposed models have two output ends; one is for breakdown detection $\hat{a}_{break}$ and the other is for next sentences prediction $\hat{a}_{sent}$.

$$\hat{a}_{break} = \mathrm{softmax}(W_{break} \cdot (o + u) + b_{\mathrm{break}}) \tag{3.7}$$

$$\hat{a}_{sent} = W_{sent} \cdot (o + u) + b_{\mathrm{sent}} \tag{3.8}$$

Thus, the numbers of elements of $\hat{a}_{break}$ and $\hat{a}_{sent}$ are, respectively, three, which is the number of breakdown labels: Not a Breakdown (NB), Possible Breakdown (PB) and Breakdown (B) as descrbed in introduction, and the sentence-embedding dimension.

4. Experiments and Results

4.1. Training Details

Training consists of dialogue breakdown detection and next-sentence prediction. Utterances and sentences used as queries are stored to external memory after each final prediction.

The loss functions to be optimized are 1) softmax cross entropy for predicting breakdown labels and 2) the mean squared error between the predicted sentence vector and a teacher signal vector. For each epoch, the set of parameters for breakdown labels is first optimized; then, that for sentence prediction is updated. We note that the set of parameters defined in (3.7) and (3.8) are, respectively, updated for the softmax cross entropy and mean squared error. The shared parameters are updated for both losses. We use the plain stochastic gradient descent (SGD) as an optimizer for both loss functions.

While several annotators may assign different labels to the same sentence of utterances, we conduct training through an instance labeled by each annotator. The sentence vectors, as teacher signals for each training epoch, are computed by the model trained in the previous training epoch.

4.2. Data Sets

We prepared data sets in both Japanese and English. The Japanese-language training data sets for all experiments consisted of ProjectNextNLP, DBDC1-dev, and DBDC2-dev data sets, with DBDC2-ref validation data sets.

The English-language training data set used for all runs was the English-language data set provided for DBDC3. No validation data set was prepared. We also prepared the following additional data sets.

Multilingual data: Multilingual training data set is prepared by simply merging two training data sets of different languages. Thus, the merged data set contained both English utterances and Japanese utterances. Mixing data sets in two languages is a kind of data augmentation under the hypothesis that some breakdowns are caused by language independent patterns of symbols such as repetition of same words and speaking random words with ignoring a question.

Data for sentence prediction: We randomly sampled passages from Wikipedia to prepared an additional training data set for the sentence prediction task. Up to 25 sentences in a passage is used. The number of passages is 1000 for DBDC3 formal runs and 5000 for experiments with convolutional attention.

4.3. Experiments for DBDC3 Formal Runs

Our experiments on the DBDC3 data set were executed using the following five models:

- English-run1: The model consisted of a plain memory network with attention over attention (plain model). The training data set contained English data only.
- Japanese-run1: Plain model with Japanese training data.
- Japanese-run2: Training to learn sentence prediction sub-task with an additional Wikipedia-sourced data set in addition to Japanese-run1.
- Japanese-run3 and English-tun2: Plain model with multilingual dialogue breakdown training data.

See [10] for detailed explanations and all results (the authors' team name is PLECO).

Results: While our character base models do not yield very high accuracy in DBDC3 formal runs, it is possible that the network structures for embedding have a negative effect. Further considerations and adjustments, such as use of recurrent neural networks, will be needed to elucidate the application of character-level embedding to breakdown detection.

By preparing additional Wikipedia-based data for the sentence prediction task, the number of characters in the training data set was increased from 2082 to 3303 in the Japanese-language case. The percentage of characters in the training data set that appeared in the test and validation data sets increased from approximately 94 to 98%. However, this increase did not yield improved accuracy.

One interesting result is that training of Japanese dialogue breakdown detection with a multilingual data set gives the model a better Jensen-Shannon (JS) divergence score than that obtained using Japanese only, as shown in Table 3. However, training of English dialogue breakdown detection using a multilingual data set does not give the model a better JS divergence score than that obtained using English only. This discrepancy may be due to differences in grammatical structures between the two languages, because English has stricter grammatical structures than Japanese, requiring stricter ordering of words and tokens, and because word sequences with loose orders may be useless for training an English detection model.

Table 3: *Jensen Shannon Divergence and Accuracy*

	JSD (ours)			Accuracy	
Run	(NB, PB, B)	(NB, PB+B)	(NB+PB, B)	ours	top
jp-1	0.1121	0.0807	0.0727	0.5146	0.6129
jp-2	0.0985	0.0698	0.0616	0.5067	
jp-3	0.0959	0.0679	0.0601	0.5386	
eng-1	0.0714	0.0427	0.535	0.2950	0.4415
eng-2	0.0774	0.0482	0.565	0.2900	

4.4. Experiments with Convolutional Attention

In this subsection, we report an accuracy comparison for the following settings:

- without_sentence_predict: AoA as the Atten, normal weighted sum as the $\mathrm{WeightOp}$ and only training for breakdown detection
- normal_atten: the models learned sentence prediction task for an additional training Wikipedia-based data set with 5000 sentences in addition to the without_sentence_predict setting
- conv_atten: $\mathrm{ConvAtten}$ in (3.4) as the Atten in addition to the normal_atten setting
- conv_atten_ws: $\mathrm{ConvWeightedSum}$ in (3.6) as the $\mathrm{WeightOp}$ in addition to the conv_atten setting

Results: The results of the experiments involving training and validation data sets in Japanese indicate that one of our approaches is effective and that incorporation of convolutional attention in the method contributes to accuracy improvement. In particular, the accuracy for both methods using convolutional attention in Figure 3 is higher than that for models with normal attention. This improvement may have been obtained because the models can correct and integrate information of local turns.

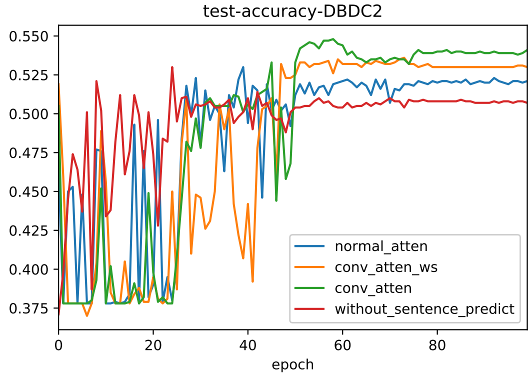

Figure 3: *Accuracy for Japanese test data set*

5. Conclusion and Future Directions

In this paper, we proposed memory-network-based dialogue breakdown detection models and methods to train breakdown detection and next utterance prediction in parallel. The used architecture made it possible to deal with character-level utterance embedding by using convolutional embedding, and improved results by using convolutional attention.

While our submission to DBDC3 did not yield a better accuracy than the top team, some remarkable findings were obtained regarding integration of local contexts in external memories, having implications for additional experiments.

Future development of our models and methods will be conducted to explore the following topics. One research direction concerns effective exploitation of two kinds of embedding: character and word level. Experiments with multi-lingual settings yielded superior JS divergence for the Japanese data. It is an advantage that character-level sentence modeling facilitates superior performance with mixing of multiple languages. Further, the degree of abstraction of the information in a sequence of symbols differs between word and character level. Therefore, the next challenge will be to combine these two levels of abstraction without losing this advantage for multi-lingual settings.

Another important direction concerns definition of *positive* and *negative meanings* of words in the context of dialogue logs, and exploitation of such meanings for breakdown detection. For instance, consider the following dialogue:

User: I bought my clothes. That's my favorite one!
System: That seems to be a low price.

Some users may think the system's response about the favorite item of clothing is negative, because the word "low price" can mean low-quality clothing, but others may not. Moreover, the word "cheap" or the term "low price" can make a positive impression in some appropriate contexts. Thus, it is important to construct models for predicting positive or negative meaning for given dialogue history contexts.

6. Acknowledgements

The authors thank to all organizers of DBDC3, Yuiko Tsunomori for her comments on explanations of formal-runs for DBDC3 data, and other Nextremer members for their supports.

7. References

[1] B. Martinovsky and D. Traum, "The error is the clue: Breakdown in human-machine interaction," *ISCA Workshop on Error Handling in Spoken Dialogue Systems*, 2003.

[2] R. Higashinaka, K. Funakoshi, Y. Kobayashi, and M. Inaba, "The Dialogue Breakdown Detection Challenge: Task Description, Datasets, and Evaluation Metrics," *Proceedings of International Conference on Language Resources and Evaluation (LREC)*, 2016.

[3] R. Higashinaka, K. Funakoshi, M. Mizukami, H. Tsukahara, Y. Kobayashi, and M. Araki "Analyzing dialogue breakdowns in chat-oriented dialogue systems, " *Interspeech Satelite Workshop, Errors by Humans and Machines in multimedia, multimodal and multilingual data processing (ERRARE 2015)*, 2015.

[4] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus "End-To-End Memory Networks," *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015.

[5] Y. Cui, Z. Chen, S. Wei, S. Wang, and T. Liu and G. Hu, "Attention-over-Attention Neural Networks for Reading Comprehension," *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

[6] S. Kobayashi, Y. Unno, and M. Fukuda, "Multi-task Learning of Recurrent Neural Network for Detecting Breakdowns of dialog and Language Modeling," *SIG-SLUD-B502-10*, 2015.

[7] T. Kubo, H. Nakayama, "Learning dialog and its breakdowns simultaneously by Neural Conversational Model," *SIG-SLUD-B505-26*, 2016.

[8] A. Bordes, Y. Boureau, and J. Weston, "Learning End-to-End Goal-Oriented Dialog," *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

[9] C.-W. Liu, R. Lowe, I.V. Serban, M. Noseworthy, L. Charlin, J. Pineau, "How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

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