Dialogue Breakdown Detection using Hierarchical Bi-Directional LSTMs

Zeying Xie, Guang Ling

Baidu Inc., Beijing, China

{xiezeying, lingguang01}@baidu.com

Abstract

In this paper, we present a hierarchical Bi-Directional LSTMs neural network designed for the dialogue breakdown detection challenge 3 (DBDC3). The task of DBDC3 is to detect inappropriate utterances that lead to dialogue breakdowns in dialogue systems. By making use of the hierarchical structure of dialogue systems, our model is context-aware and can be trained directly from word sequences to breakdown labels in an end-to-end manner. Thus our model is generic and requires no feature engineering, making it applicable to different dialogue systems. We evaluate our model on 4 dialogue systems provided by the DBDC3 datasets. Experiment results show that our proposed model achieves high accuracy, and outperforms the CRF baseline in terms of both classification-related and distribution-related metrics without careful fine-tuning.

Index Terms: Bi-LSTM, hierarchical architecture, end-to-end, dialogue breakdown detection, dialogue systems

1. Introduction

With the increasing pervasiveness of smart phones and smart devices, spoken dialogue systems are gaining ever growing attention from both academic and industry. Spoken dialogue system is considered as a candidate for next generation human-machine interface. A lot of spoken dialogue system based assistants have emerged, including Siri, Google Assistant, Amazon Echo, Cortana and Xiaoice. The literature of dialogue system research can be broadly classified into two categories, one that is aimed at helping user to gain knowledge and providing useful services and one that can chat with users without completing any specific tasks. The former one is usually called task-oriented or goal-oriented dialogue system [1] and the later chat-oriented dialogue system [2, 3]. Although dialogue systems are improving substantially, the user experience of such systems are still unsatisfactory. The systems fail to understand the intention of the users' utterance and respond inappropriately occasionally. We call this dialogue breakdowns [4] and detection of them is crucial to improve user experience [5]. This paper focuses on chat-oriented dialogue breakdowns.

We now briefly introduce chat-oriented dialogue system and point out the difficulties inherent to the dialogue system problem that could result in breakdowns. Chat-oriented dialogue system provides the ability to chat with user, mimic the conversation between two people. The objective of chat-oriented dialogue system is to respond properly and convincingly to users' utterances. Recent chat-oriented dialogue systems usually adopt neural machine translation architecture [6, 7] and are trained in an end-to-end manner.

The design and implementation of dialogue system has evolved from labor-intensive rule-based systems [8] to data driven approaches [1, 2, 3, 9]. Recent advance of deep learning has inspired many applications of neural models to dialogue systems. Both selection based [3, 10] and generation based methods [2, 11] have been proposed to build chat-oriented dialogue systems.

The user experience of spoken dialogue systems is still far from satisfactory. The system usually fails to understand the intention of the user, especially during a multi-turn conversation. When the systems fail to understand the intention of a user, it will produce response based on its false understanding, which could result in obviously irrelevant and inappropriate responses. The user experience would be unpleasant at least under such circumstances. If we could detect such system breakdowns, i.e. that the system is producing irrelevant or inappropriate responses, we could take precautions and ask the user to reformulate his/her questions.

The causes of breakdowns in current dialogue systems are multi-aspects. In chat-oriented dialogue systems, take generation based system for example, the encoder could fail to encode an utterance correctly and the decoder could produce irrelevant responses. Variability and ambiguity of natural language also cause difficulties in understanding. Often the true meaning can only be inferred in a given context. In addition, in spoken dialogue system, the input is often the transcription produced by automatic speech recognition module, which may produce erroneous sentences.

Detect the dialogue breakdowns and handle them properly could be a valuable alternative way to build better dialogue system because the afore-mentioned causes for unsatisfactory dialogue system performance is unlikely to be solved very soon. In this paper, we propose a novel hierarchical Bi-LSTM based method to detect the system breakdowns in an end-to-end manner. In section 2 we describe the dialogue breakdown detection task. We present the model in detail in section 3. The emperical analysis is conducted in section 4 and we conclude our method in section 5.

2. Task Description

The dialogue system breakdown is defined as a situation in a dialogue where users cannot proceed with the conversation. The task of Dialogue Breakdown Detection Challenge 3 (DBDC3) [12, 13] is to detect whether the system utterance causes dialogue breakdowns. The developed dialogue breakdown detector is required to output both a dialogue breakdown label and a distribution of these labels. Due to the subjective nature of deciding whether the user can proceed with the conversation, the states of system breakdowns includes the following three labels:

- **Not a breakdown (NB)**: The conversation can continue easily.
- **Possible breakdown (PB)**: It is difficult to continue the conversation smoothly.
- **Breakdown (B)**: It is difficult to continue the conversation.

Table 1: *Statistics of English datasets*

	TKTK		IRIS		CIC		YI	
	train	test	train	test	train	test	train	test
No. of sessions	100	50	100	50	115	50	100	50
No. of annotators	30	30	30	30	30	30	30	30
NB (Not a Breakdown)	35.1%	44.3%	32.9%	34.5%	28.9%	29.1%	34.8%	35.4%
PB (Possible Breakdown)	27.6%	29.2%	27.8%	29.3%	29.8%	39.3%	36.1%	40.3%
B (Breakdown)	37.3%	26.5%	39.4%	36.2%	41.3%	31.6%	29.1%	24.3%

2.1. Datasets

DBDC3 distributed multi-turn human-system dialogue session datasets along with human annotated breakdown labels. Eight session datasets from different chat-oriented dialogue systems are available, among which four are English datasets and four are Japanese datasets. In this paper, we only focus on English datasets.

The four English datasets are: TKTK, IRIS, CIC and YI. Each dataset is separated into training data for model development and test data for model testing. All dialogue sessions are 20 or 21 utterances long and include 10 system responses. Table 1 summarizes the statistics of the English datasets.

2.2. Evaluation Metrics

DBDC3 uses two types of evaluation metrics: classification-related metrics and distribution-related metrics.

2.2.1. Classification-related Metrics

classification-related metrics are evaluated on breakdown labels predicted by model against the gold labels determined by majority voting of human annotations. The metrics used by DBDC3 are as follows:

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

2.2.2. Distribution-related Metrics

Distribution-related metrics evaluate the similarity of the predicted and gold breakdown distributions. The metrics used by DBDC3 are as follows:

- 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.

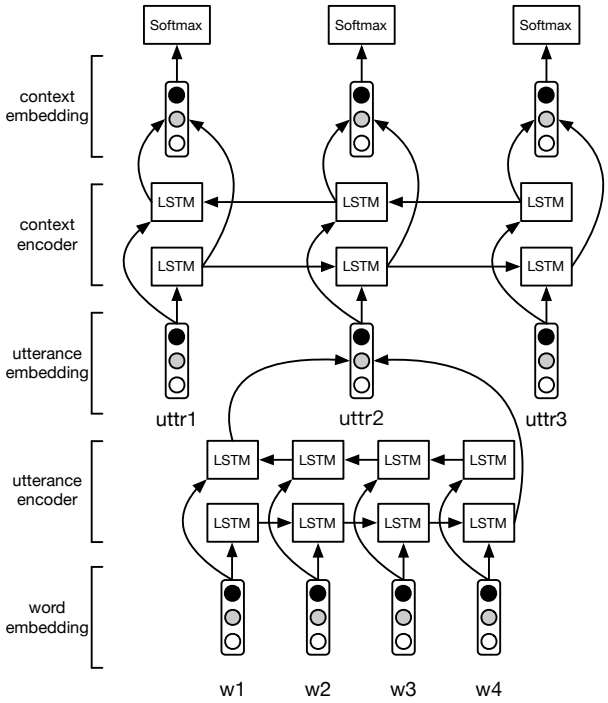

Figure 1: *Hierarchical Bi-Directional LSTMs model architecture.*

- Mean Squared Error (NB+PB,B): Mean squared error when NB and PB are regarded as a single label.

However, the results may not be as easily interpretable as the classication-related metrics because they do not directly translate to detection performance.

3. Proposed Model

In this section, we describe the motivation and components of Hierarchical Bi-Directional LSTMs (H-Bi-LSTM) neural network in detail. The overall architecture of H-Bi-LSTM model is shown in Figure 1.

3.1. Motivation

A well-known and effective neural network called LSTM is designed to model sequence dependence and has achieved state-of-the-art in many NLP tasks. As a dialogue utterance is composed by sequence of words, we use a Bi-LSTM to encode a utterance by its corresponding words. Dialogue system interacts with user in sequential order, and every system utterance is generated by considering the history of the dialogue up the

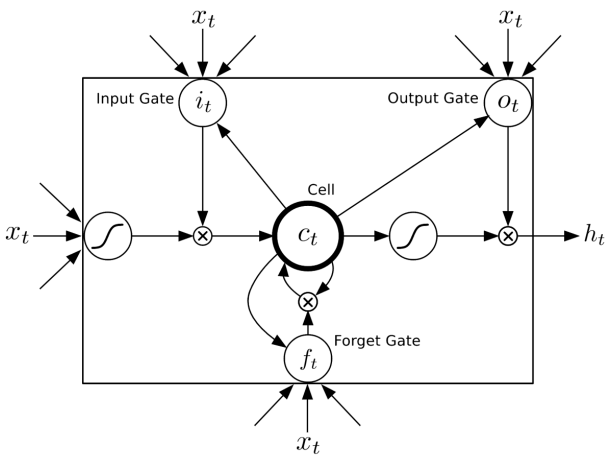

Figure 2: *Long Short-Term Memory Unit Structure.*

latest turn. It is natural to abstract the process of dialogue interactions as a sequence model. Thus we use another Bi-LSTM over the utterance encoder for dialogue context encoding. The architecture of our model mimics the hierarchical structure of dialogue, and results in the name of the model. Besides, Our model is designed to be truly end-to-end so that it can be easily applied to different dialogue systems without modification, no feature engineering or extra task-specific resources are required.

3.2. Bi-Directional LSTM

Recurrent Neural Network (RNN) is a powerful model that makes use of sequential information. Unfortunately, in practice standard RNN often fails to capture long term dependencies due to gradient vanishing/exploding [14, 15]. Numerous variants of RNN are proposed to address this problem for RNN, among which LSTM [16] is proved to work amazingly well and is widely applied to various real-world problems.

3.2.1. LSTM Unit

Basically, a LSTM unit consist of a memory cell and three multiplicative gates: input gate, forget gate and output gate, which control the proportions of information to flow into or out of the memory. Figure 2 gives the basic structure of an LSTM unit.

Formally, the formulas to update an LSTM unit at time t are given by:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{h}_{t-1} + \mathbf{U}_i\mathbf{x}_t + \mathbf{b}_i) \quad (1)$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{h}_{t-1} + \mathbf{U}_f\mathbf{x}_t + \mathbf{b}_f) \quad (2)$$
$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c\mathbf{h}_{t-1} + \mathbf{U}_c\mathbf{x}_t + \mathbf{b}_c) \quad (3)$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \quad (4)$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{h}_{t-1} + \mathbf{U}_o\mathbf{x}_t + \mathbf{b}_o) \quad (5)$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (6)$$

where σ denotes sigmoid function and $\odot$ denotes element-wise product. $\mathbf{x}_t$ is the input vector at time t, $\mathbf{h}_t$ is hidden state vector at time t. $\mathbf{U}_i, \mathbf{U}_f, \mathbf{U}_o, \mathbf{U}_c$ denote weight matrices of input $\mathbf{x}_t$, $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o, \mathbf{U}_c$ denote weight matrices of hidden state $\mathbf{h}_t$, $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o, \mathbf{b}_c$ denote the corresponding bias vectors.

3.2.2. Bi-LSTM

For many sequence labelling tasks it is beneficial to have access to both past and future information. A standard LSTM only knows context of the past and nothing about the future. Bi-directional LSTM (Bi-LSTM) [17] offers an elegant solution to this problem, and has been proven to outperform unidirectional LSTM consistently by previous work. The basic idea is to present each sequence forwards and backwards to two separate recurrent hidden layers to capture past and future information, respectively. Then the two hidden states are concatenated to form the final output. The output at time t of Bi-LSTM is calculated as follows:

$$\overrightarrow{\mathbf{h}}_t = \overrightarrow{\text{LSTM}}(\mathbf{x}_t) \quad (7)$$
$$\overleftarrow{\mathbf{h}}_t = \overleftarrow{\text{LSTM}}(\mathbf{x}_t) \quad (8)$$
$$\mathbf{h}_t = \overrightarrow{\mathbf{h}}_t \oplus \overleftarrow{\mathbf{h}}_t \quad (9)$$

where $\oplus$ denotes vector concatenation, $\overrightarrow{\text{LSTM}}, \overleftarrow{\text{LSTM}}$ denote the forward LSTM and backward LSTM, respectively. $\overrightarrow{\mathbf{h}}_t$, $\overleftarrow{\mathbf{h}}_t$ denote the corresponding forward and backward outputs.

3.3. Model Architecture

H-Bi-LSTM consists of components that extract semantic representations of dialogues from low-level to high-level: word embedding, utterance encoder and context encoder. Context representations are then fed into a fully-connected layer and with softmax activation to output probabilities over possible breakdown labels.

3.3.1. Word Representation

Each word in the vocabulary is represented as a fixed-length semantic vector through an embedding matrix $\mathbf{W}_w$. Due to the limited training data of DBDC3, We use the pre-trained GloVe embeddings [18] for word representation to prevent overfitting.

3.3.2. Utterance Representation

We use a Bi-LSTM for utterance encoding. Given an utterance with N words $[\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N]$, we first embed the words through $\mathbf{W}_w$ to get word embeddings $[\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_N]$. Word embeddings of the utterance are then fed into the Bi-LSTM encoder as inputs and the final output $\mathbf{h}_N$ is regarded as the corresponding utterance embedding $\mathbf{uttr}$. User utterance embedding and system utterance embedding of the same dialogue turn are concatenated together.

3.3.3. Context Representation

Similarly, we use another Bi-LSTM as dialogue context encoder. Given the utterance representations of a dialogue with M turns $[\mathbf{uttr}_1; \mathbf{uttr}_2; \ldots; \mathbf{uttr}_M]$, we apply the encoder over them to get the context representations of every turn $[\mathbf{c}_1; \mathbf{c}_2; \ldots; \mathbf{c}_M]$.

3.3.4. Breakdown Detection

The context embedding $\mathbf{c}$ can be used as high-level features to calculate dialogue breakdown scores, $\mathbf{w}_b, \mathbf{b}_b$ are the corresponding weight matrix and bias, the scores are then translated to probabilities via a softmax normalization:

$$\hat{\mathbf{p}} = \text{softmax}(\mathbf{W}_b\mathbf{c} + \mathbf{b}_b) \quad (10)$$

Table 2: *Evaluation results on DBDC3 test datasets (English).*

Model	Accuracy	F1 (B)	F1 (PB+B)	JSD(NB, PB,B)	JSD(NB, PB+B)	JSD(NB+ PB,B)	MSE(NB, PB,B)	MSE(NB, PB+B)	MSE(NB+ PB,B)
CRF Baseline	0.4285	0.3543	0.7622	0.4409	0.2687	0.2985	0.2185	0.2171	0.2578
Majority Baseline	0.3720	0.3343	**0.8927**	**0.0393**	0.0237	0.0257	**0.0224**	0.0278	0.0264
PLECO run1	0.2950	**0.3636**	0.8744	0.0714	0.0427	0.0535	0.0415	0.0455	0.0632
RSL17BD run2	0.4310	0.3201	0.8400	0.0412	0.0256	**0.0225**	0.0241	0.0301	**0.0246**
NCDS run1	0.3605	0.2076	0.3458	0.0412	0.0248	0.0254	0.0237	0.0287	0.0270
KTH run1	0.3375	0.3487	0.8423	0.4445	0.2343	0.2058	0.2240	0.1752	0.1476
SAM2017 run1	0.4060	0.2413	0.2160	0.2823	0.2377	0.0805	0.1441	0.2652	0.0621
H-Bi-LSTM #1 (ours run1)	0.4295	0.3210	0.7627	0.0807	0.0438	0.0444	0.0471	0.0501	0.0497
H-Bi-LSTM #2 (offline test)	**0.4595**	0.3631	0.8049	**0.0393**	**0.0231**	0.0250	0.0228	**0.0270**	0.0276

3.3.5. Loss

We use cross entropy of the predicted distributions $\hat{\mathbf{p}}$ with respect to true distributions $\mathbf{p}$ as the training loss:

$$L = -\sum_i p_i \log(\hat{p}_i) \tag{11}$$

4. Experiment

In this section, we evaluate the effectiveness of our model on the four English dialogue session datasets as described in section 2.

4.1. Data Preparing

Before feeding data to our model, we perform the following preparations:

- Each dialogue utterance is tokenized into words using Stanfords CoreNLP toolkit [19] and words are converted to lower case to keep consistent with the vocabulary of GloVe word embeddings. Out-of-vocabulary words are replaced with a special token ⟨**UNK**⟩. Utterances with more than 50 words are truncated.
- Human annotated breakdown labels of every system utterance are converted to probabilities, which will be regarded as the true distribution when calculating model loss.
- For each dialogue dataset, we randomly split the training data: 80% of data are used for training and the remaining 20% are used for validation.

4.2. Model Training

We use the same hyper-parameters which are determined by a rough grid search for all the experiments. For word embeddings, we use the public available GloVe 100-dimensional embeddings trained on 6 billion words from Wikipedia and web text. The dimension of both Bi-LSTM hidden states are 50. To mitigate overtting, we apply dropout [20] layers on word embeddings, utterance embeddings and context embeddings respectively with 0.2 dropout probability. We choose Adam optimizer to minimize the training loss with an initial learning rate of 0.001 and batch size of 1. The model is trained 50 epochs for parameter optimization.

4.2.1. True Distribution Setting

As mentioned above, we use cross entropy between distribution predicted by our model and true distribution. In our experiments, there are two settings for calculating true label distribution:

1. The probability of the most frequent dialogue breakdown label is set to 1.0, and probabilities of other labels are set to 0. (used in H-Bi-LSTM #1)
2. The true distribution is calculated by annotation counts of each breakdown label divided by total annotation counts. (used in H-Bi-LSTM #2)

For example, suppose a system dialogue turn is annotated with NB:PB:B=1:3:4. In setting 1, the true distribution is [0, 0, 1]; In setting 2, the true distribution is [0.125, 0.375, 0.5].

H-Bi-LSTM models using true distribution setting 1 and 2 are named H-Bi-LSTM #1 and H-Bi-LSTM #2, respectively. The results we submitted was generated by H-Bi-LSTM #1. After the reference annotations of the test data were released, we used H-Bi-LSTM #2 to do extra offline experiments.

4.3. Baselines

We compared H-Bi-LSTM with two baselines provided by DBDC3: One is a 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. The other one is simply a majority baseline which outputs only the most frequent dialogue breakdown label in the development set with averaged probability distributions.

4.4. Results and Discussion

The final results of our proposed model along with two baselines and five other teams are shown in Table 2. We only submitted one run of our model results(denoted as H-Bi-LSTM #1). As can be seen from the table, H-Bi-LSTM #1 achieves a high accuracy that outperforms the majority baseline by 15%, and improves the distribution metrics by a huge margin comparing to the CRF baseline. However, H-Bi-LSTM #1 is not performing good enough comparing to other teams, especially in turns of F1 scores and all distribution-related metrics.

After carefully analyzing the results of H-Bi-LSTM #1, we figure out the performance of H-Bi-LSTM #1 is badly hurt by its true distribution setting. Thus, we set up extra offline experiments(H-Bi-LSTM #2), results show that the performance of our model is significantly improved. H-Bi-LSTM

#2 improves all the metrics comparing to H-Bi-LSTM #1 by a huge margin, and outperforms all teams on 4 metrics. Besides, F1(B), JSD(NB+PB,B) and MSE(NB,PB,B) of H-Bi-LSTM #2 are very closed to the best results. As from the table, teams with good results in the classication-related metrics did not perform as well in distribution-related metrics and vice versa, while H-Bi-LSTM #2 can achieve good results in both metric categories.

The experiment results demonstrate the effectiveness and robustness of our proposed model. However, there is still some room for improvement comparing to the majority baseline, either by hyper-parameters fine-tuning or model ensemble.

5. Conclusion

In this paper, we presents our proposed dialogue breakdown detector for DBDC3. The model we used is in a novel hierarchical Bi-LSTM neural network that can capture semantic representations of dialogues progressively from words and utterances. Submitted results on DBDC3 test datasets demonstrate that our model achieves a high accuracy, and outperforms the CRF baselines in several evaluation metrics. Extra offline experiments showed that the performance of our model can be significantly improved by using another true distribution setting. In the future work, attention mechanism [21] can be integrated into H-Bi-LSTM to help generate better representations, we believe this will further improve the performance of the model.

6. Acknowledgement

This paper is supported by National Basic Research Program of China (973 program No. 2014CB340505).

7. References

[1] J. D. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech & Language*, vol. 21, no. 2, pp. 393–422, 2007.

[2] O. Vinyals and Q. V. Le, "A neural conversational model," *CoRR*, vol. abs/1506.05869, 2015. [Online]. Available: http://arxiv.org/abs/1506.05869

[3] X. Zhou, D. Dong, H. Wu, S. Zhao, D. Yu, H. Tian, X. Liu, and R. Yan, "Multi-view response selection for human-computer conversation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2016, pp. 372–381. [Online]. Available: http://aclweb.org/anthology/D/D16/D16-1036.pdf

[4] B. Martinovsky and D. Traum, "The error is the clue: Breakdown in human-machine interaction," UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY CA INST FOR CREATIVE TECHNOLOGIES, Tech. Rep., 2006.

[5] R. Higashinaka, K. Funakoshi, Y. Kobayashi, and M. Inaba, "The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*, 2016. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2016/summaries/525.html

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[7] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 2015, pp. 1412–1421. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1166.pdf

[8] J. Cassell, C. Pelachaud, N. I. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone, "Animated conversation: rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents," in *Proceedings of the 21th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1994, Orlando, FL, USA, July 24-29, 1994*, 1994, pp. 413–420. [Online]. Available: http://doi.acm.org/10.1145/192161.192272

[9] M. Henderson, "Discriminative Methods for Statistical Spoken Dialogue Systems," Ph.D. dissertation, University of Cambridge, 2015.

[10] Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li, "Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 2017, pp. 496–505. [Online]. Available: https://doi.org/10.18653/v1/P17-1046

[11] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A persona-based neural conversation model," *CoRR*, vol. abs/1603.06155, 2016. [Online]. Available: http://arxiv.org/abs/1603.06155

[12] R. Higashinaka, K. Funakoshi, M. Inaba, Y. Tsunomori, T. Takahashi, and N. Kaji, "Overview of dialogue breakdown detection challenge 3," in *Proceedings of Dialog System Technology Challenge 6 (DSTC6) Workshop*, 2017.

[13] R. Higashinaka, K. Funakoshi, Y. Kobayashi, and M. Inaba, "The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics." in *LREC*, 2016.

[14] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[15] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies."

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[19] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. [Online]. Available: http://www.aclweb.org/anthology/P/P14/P14-5010

[20] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.