

# Datasets and Benchmarks for Task-Oriented Log Dialogue Ranking Task

Xinnuo Xu<sup>1</sup>, Yizhe Zhang<sup>2</sup>, Lars Liden<sup>2</sup>, Sungjin Lee<sup>3\*</sup>

<sup>1</sup>The Interaction Lab, Heriot-Watt University, Edinburgh

<sup>2</sup>Microsoft Research, Redmond, WA, USA

<sup>3</sup>Amazon Alexa AI, Redmond, WA, USA

xx6@hw.ac.uk, Yizhe.Zhang, Lars.Liden@microsoft.com, sungjinl@amazon.com

## Abstract

Although the data-driven approaches of some recent bot building platforms make it possible for a wide range of users to easily create dialogue systems, those platforms don't offer tools for quickly identifying which log dialogues contain problems. Thus, in this paper, we (1) introduce a new task, log dialogue ranking, where the ranker places problematic dialogues higher (2) provide a collection of human-bot conversations in the restaurant inquiry task labelled with dialogue quality for ranker training and evaluation (3) present a detailed description of the data collection pipeline, which is entirely based on crowd-sourcing (4) finally report a benchmark result of dialogue ranking, which shows the usability of the data and sets a baseline for future studies.

**Index Terms:** dialogue ranking, dialogue quality, language resource, dialogue system

## 1. Introduction

Task-oriented dialogue systems provide a natural interface to accomplish various daily-life tasks such as restaurant finding and flight booking. Data-driven approaches offered by common bot building platforms (e.g. Google Dialogflow, Amazon Alexa Skills Kit, Microsoft Bot Framework) make it possible for a wide range of users to easily create dialogue systems with a limited amount of data in their domain of interest.

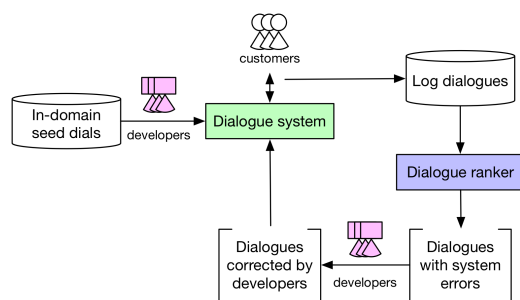


Figure 1: Operational loop of dialogue system development.

Typically, the development process of a dialogue system based on data-driven approaches [1, 2] goes around an operational loop in Fig 1: (1) The cycle begins with a developer creating a training dataset with seed dialogues. (2) A dialogue system is trained and deployed. (3) Real users interact with the system and generate log dialogues. (4) The developer reviews the logs to identify which log dialogues contain problems. (5) The developer updates the training dataset to fix the problems. (6) The cycle repeats from step 2). Of all steps, (4) is the most

\* The work was done while Sungjin was still at Microsoft Research

significant in slowing down the loop, because of the sheer volume of log dialogues that can be generated and the need to manually inspect each. Thus, it is essential to support tools that help developers quickly identify problematic log dialogues.

To achieve this goal, we introduce a new task, task-oriented log dialogue ranking whose goal is to place problematic dialogues higher in the rank trained on a small number of labeled data. There are prior studies on dialogue quality assessment. [3, 4] investigate automatic ways of dialogue evaluation. [5, 6, 7, 8, 9] focus on dialogue breakdown detection during the dialogue. [10, 11] estimate dialogue quality to improve dialogue policy. However, none has directly dealt with ranking in terms of the quality of task-based dialogues. For automatic error handling, [12, 13, 14] examined different types of errors. Whereas, for our task, it suffices to rank problematic dialogues higher than normal ones so that developers may take appropriate measures based on the goal of their bots, e.g., adding new slots or system actions to handle previously uncovered scenarios.

In the rest of this paper, we present a crowd-sourcing based data collection pipeline using Amazon Mechanical Turk (AMT) in Section 2 and describe the collected data in Section 3. To demonstrate the utility of the data collected, we report a benchmark result for the dialogue ranking task in Section 4 and 5.

## 2. Data Collection Set-up

To collect log dialogues, we deployed the Pydial restaurant bot<sup>1</sup> via the Amazon Mechanical Turk (AMT) platform.<sup>2</sup> The task for the bot is to search for restaurants based on user's requirements in a multi-turn natural language communication. We ask turkers to find restaurants that satisfy automatically generated requirements, such as food type, location and price range, by chatting with the restaurant bot. Figure 2 shows an example task.

To make the conversation natural and keep users engaged, we encourage turkers to speak in natural utterances by 1) detecting one token user utterance and asking the turker to rephrase the answer with a natural sentence, 2) closing the AMT session and eliminating the conversation if the turker refuses to use natural response for three times. To increase the diversity of the dialogue collection we do not allow any turkers to carry out more than 20 dialogues in total.

At the end of each task, turkers are required to answer a questionnaire including two questions, 1) whether they found restaurants satisfying their requirements, 2) label contextually unnatural turns in the conversation they experienced. The interface of the questionnaire is shown in Figure 3. To control the quality of turker's judgements we (1) ask turkers to rephrase

<sup>1</sup><http://www.camdial.org/pydial/>

<sup>2</sup>We use the data collection toolkit offered by ParlAI [http://www.parl.ai/static/docs/tutorial\\_mturk.html](http://www.parl.ai/static/docs/tutorial_mturk.html).

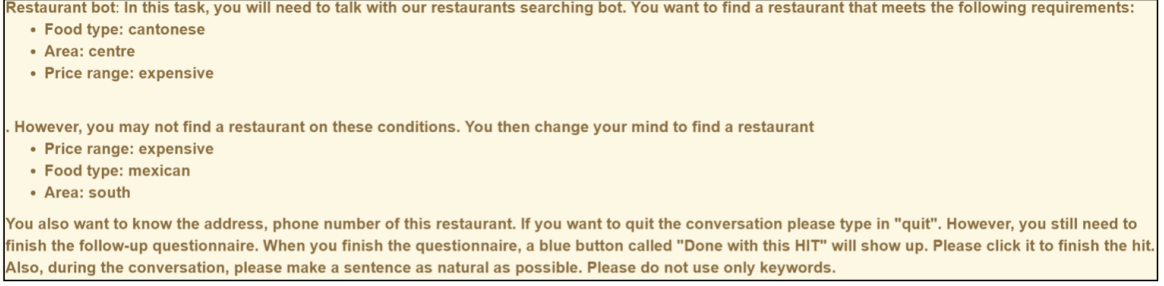


Figure 2: The opening interface of the collection task.

their feedback if their judgements follow incorrect formats, (2) automatically collect unnatural turns in which the bot fails to understand the turker’s utterance and require turkers to regenerate their judgements that miss the seed unnatural turns, (3) close the AMT session and eliminate the conversation if the turker refuses to offer qualified judgements for more than three times.

### 3. Dataset Description

From the collected dialogues, we label successful dialogues without any contextually unnatural turns as 0 (normal dialogue), and the rest as 1 (problematic dialogue). In general, we collected 1470 normal dialogues and 1896 problematic dialogues, with average dialogue length as 5.87 turns, 8.28 turns respectively. Examples are shown in Table 1.

Fig 4, 5, 6 show the distribution of the number of sentences, number of tokens in user utterances, utterance repetition rate of the problematic and normal dialogues respectively. The repetition rate is calculated per dialogue by  $1 - N^u/N$ , where  $N^u$  and  $N$  denote the number of unique sentences and the number of sentences in a dialogue respectively. It is obvious that normal dialogues are more efficient (shorter dialogue length and lower repetition rate) with higher user engagement (longer user utterances). We randomly split the corpus into train/validation/test sets (normal dialogues 1321/100/200; problematic dialogues 1029/100/200).

### 4. Benchmarks

A dialogue ranker aims to assign higher scores to problematic dialogues than normal ones so that developers may quickly identify problematic dialogues in the ranked list of log dialogues. Formally, given a dialogue  $\mathbf{D}$  where,

$$\mathbf{D} = \{S_1, U_1, S_2, \dots, U_{n-1}, S_n\}$$

a dialogue ranker  $\psi$  produces a score of  $\mathbf{D}$  being problematic where  $S_i$  and  $U_i$  are the system and user utterance in  $i^{th}$  turn, respectively.<sup>3</sup> To train the dialogue ranker  $\psi$ , we formulate the ranking task as binary classification where problematic and normal dialogues correspond to positive and negative classes, respectively. We optimize the cross-entropy objective:

$$\mathcal{L}_{\text{xent}} = \frac{1}{N} \sum_{i=1}^N [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$$

where  $\hat{y}_i = 1/(1 + \exp(-\psi(\mathbf{D}_i)))$  and  $y_i = 1$  for positive class and 0 otherwise. We apply two document classification models for the dialogue ranking task.

<sup>3</sup>One turn consists of a pair of system and user utterances.

#### 4.1. BiLSTM+MLP:

Fig 7 shows the structure of the BiLSTM+MLP. We first use a bidirectional Long short-term memory (Bi-LSTM) [15] to encode utterances in dialogue  $\mathbf{D}$ :

$$\mathbf{D}^{emb} = \{S_1^{emb}, U_1^{emb}, S_2^{emb}, \dots, U_{n-1}^{emb}, S_n^{emb}\}$$

where  $S_i^{emb} = Bi-LSTM(S_i)$  and  $U_i^{emb} = Bi-LSTM(U_i)$ . After that, we apply a Feed-Forward Network (FFN) followed by a sigmoid activation to each pair of system-user utterances to measure the consistency of adjacency pairs:

$$A_i = \text{sigmoid}(FFN_{pair}([X_i^{emb}, Y_i^{emb}])) \quad (1)$$

where  $(X_i^{emb}, Y_i^{emb})$  is either  $(S_i^{emb}, U_i^{emb})$  or  $(U_i^{emb}, S_{i+1}^{emb})$ . Finally, the ranker  $\psi$  produces a ranking score for the dialogue based on the consistency scores:

$$\psi(\mathbf{D}) = \text{sigmoid}(FFN_{pred}([A_1, \dots, A_{n-1}])) \quad (2)$$

Note that the FFNs in Eq 1 and Eq 2 are using two different sets of parameters.

#### 4.2. HierAttn:

Inspired by [16], we adopt Hierarchical Attention (HierAttn) for the dialogue ranking. Fig 8 shows the structure for the HierAttn. We first use a Bi-LSTM to encode each utterance. Then the embedding for an utterance is calculated as a weighted sum of the contextual word representations. For example, the embedding  $S_i^{emb}$  for the system utterance in the  $i$ th turn is:

$$u_{it} = \tanh(W_w h_{it} + b_w) \quad (3)$$

$$\alpha_{it} = \frac{\exp(u_{it}^\top u_w)}{\sum_j \exp(u_{ij}^\top u_w)} \quad (4)$$

$$S_i^{emb} = \sum_t \alpha_{it} h_{it} \quad (5)$$

where  $h_{it} = [\vec{h}_{it}, \overleftarrow{h}_{it}]$  is the word annotations, a concatenation of the hidden states of the Bi-LSTM.  $W_w$  and  $b_w$  are parameters for the non-linear transition. Word context vector  $u_w$  is randomly initialized and jointly learned during training.

We introduce the same structure with different parameters to map the utterance embeddings to a document embedding:

$$u_i = \tanh(W_s h_i + b_s) \quad (6)$$

$$\alpha_i = \frac{\exp(u_i^\top u_s)}{\sum_j \exp(u_j^\top u_s)} \quad (7)$$

$$v = \sum_i \alpha_i h_i \quad (8)$$

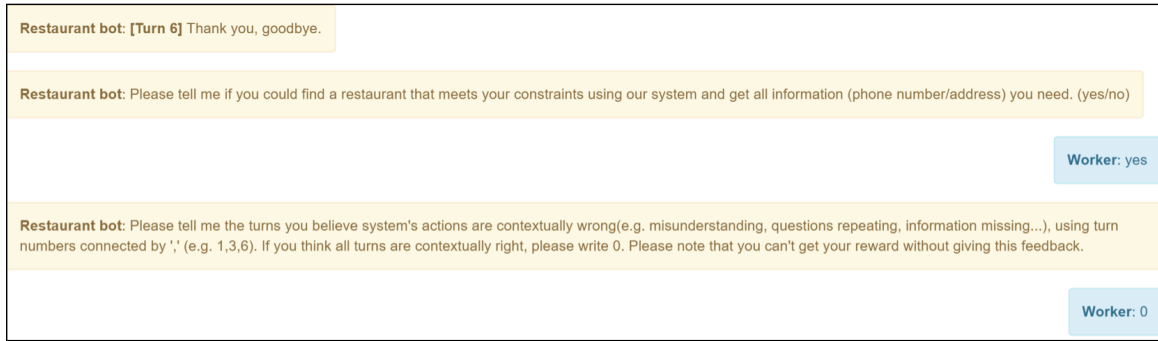


Figure 3: The ending interface of the collection task.

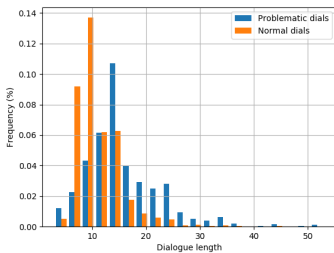


Figure 4: Average length of dialogues in problematic and normal set.

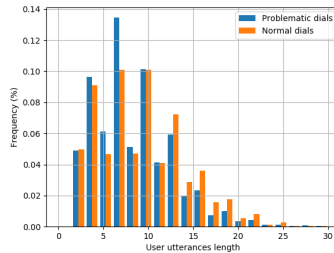


Figure 5: Average length of user utterances in problematic and normal set.

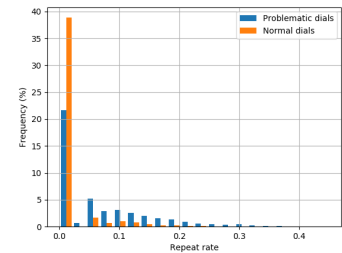


Figure 6: Repeat rate of dialogues in problematic and normal set.

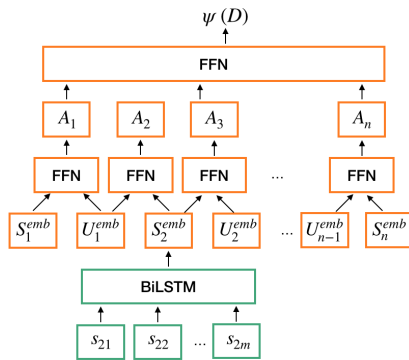


Figure 7: Structure for BiLSTM+MLP.

where the document embedding  $v$  is the weighted sum of the contextual utterance representations that is a concatenation of the hidden states of the sentence-level Bi-LSTM.  $W_s$  and  $b_s$  are parameters for the non-linear transition and  $u_s$  is the document context vector.  $\psi(\mathbf{D})$  is calculated as  $\text{sigmoid}(FFN(v))$ .

## 5. Experiments and results

We test the two dialogue rankers on the 400-instance balanced test set. To get stable and convincing results, we randomly sample balanced training sets (1000 problematic dialogues and 1000 normal dialogues) for 10 times and train dialogue rankers on each set. Then we report average performance for each ranker over 10 models.

For both models, the vocabulary size is 150 and the dimension of word embeddings is 64. All bidirectional RNNs are one layer LSTM of size 64. Adam optimizer [17] is used to update the parameters with a learning rate 0.001. BiLSTM+MLP

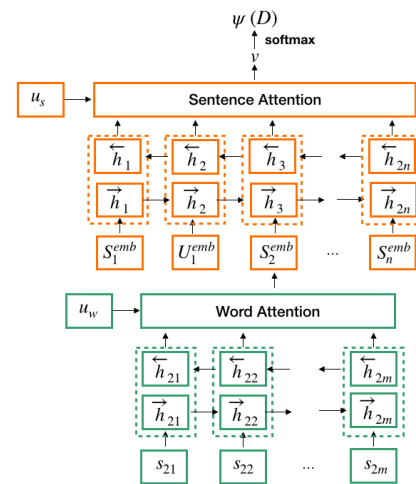


Figure 8: Structure for Hierarchical Attention.

is using the publicly available AllenNLP<sup>4</sup> framework [18]. HierAttn is implemented on Keras<sup>5</sup>.

We use ranking metrics Precision at  $k$  ( $P@K$ ) for evaluation, which corresponds to the number of problematic dialogues in the top  $k$  ranked options.

An alternative avenue can be to leverage off-the-shelf dialogue quality classifiers which are trained on open-domain corpora such as dialogue breakdown detection challenge systems in DSTC6 [19].<sup>6</sup> In fact, if off-the-shelf models work

<sup>4</sup><https://allennlp.org>

<sup>5</sup>The HierAttn is based on code <https://github.com/richliao/textClassifier>

<sup>6</sup><https://github.com/dbd-challenge/dbdc3>

Table 1: *One normal and one problematic examples in the labelled corpus.*

<b>One normal example in the labelled corpus</b>	
<b>SYS:</b>	hello, welcome to the restaurant finding system. how may i help you?
<b>USER:</b>	hi i ' m looking for a restaurant.
<b>SYS:</b>	what part of town do you have in mind?
<b>USER:</b>	the east part of town please.
<b>SYS:</b>	what kind of food would you like?
<b>USER:</b>	i'd like chinese food please.
<b>SYS:</b>	yu garden is a nice place, it serves chinese food and it is in the east part of town.
<b>USER:</b>	what is the phone number?
<b>SYS:</b>	yu garden is a nice place their phone number is 0 1223 248882.
<b>USER:</b>	what is the address?
<b>SYS:</b>	yu garden is a nice place their address is 529 newmarket road fen ditton.
<b>USER:</b>	thank you
<b>SYS:</b>	thank you, goodbye.
<b>One problematic example in the labelled corpus</b>	
<b>SYS:</b>	hello, welcome to the restaurant finding system. how may i help you?
<b>USER:</b>	hello i would really like to eat some cheap indian tonight
<b>SYS:</b>	what part of town do you have in mind?
<b>USER:</b>	i would really like to eat in the centre of town
<b>SYS:</b>	i am sorry but there is no place, in the expensive price range, serving corsica food and in the area of centre.
<b>USER:</b>	i would really like so indian thats cheap in the centre of town
<b>SYS:</b>	the gandhi is a nice place, it is in the cheap price range, it is in the centre of town and it serves indian food.
<b>USER:</b>	thats great, have you got a contact number?
<b>SYS:</b>	what part of town do you have in mind?
<b>USER:</b>	the centre of town
<b>SYS:</b>	efes restaurant is a nice place, it is in the centre of town, it is in the moderate price range and it serves turkish food.
<b>USER:</b>	i want to eat at the gandhi have you got the address?
<b>SYS:</b>	the gandhi is a nice place their address is 72 regent street city centre.
<b>USER:</b>	thank you, have you got a phone number for the gandhi?
<b>SYS:</b>	the gandhi is a nice place their phone number is 0 1223 353942.
<b>USER:</b>	thanks
<b>SYS:</b>	chiquito restaurant bar is a nice place, it serves mexican food, it is in the expensive price range and it is in the south part of town.
<b>USER:</b>	quit
<b>SYS:</b>	thank you, goodbye.

well without requiring any domain-specific data, one would not have to collect domain-specific data to train ranking models. Since the DBDC3 model operates at the utterance level, we aggregate the utterance scores to obtain a dialogue-level score. Based on our experiments, we choose to use a model trained on the collection of four challenge datasets (CIC\_115, IRIS\_100, TKTK\_100, YL\_100) and aggregate the utterance scores through summation.

### 5.1. Results and Discussion

- *DBDC3 vs. Supervised models:* Table 2 shows that the ranker that is trained on existing open-domain corpora does not work as well as the models specifically trained on a task-oriented corpus, indicating that the use of domain-relevant data is crucial in obtaining high performance.
- *BiLSTM+MLP vs. HierAttn:* Table 2 shows that except for the P@30, HierAttn is either same with or outperforms BiLSTM+MLP. Both models achieve over 90% precision in top ranked dialogues (top50). It approves that rankers trained on the small dialogue set collected by the pipeline described in the Section 2 are helpful for developers to discriminate problematic dialogues in practical. However, the accuracy drops when  $k$  is over

100. It shows the challenges of the task-oriented dialogue ranking task.

Table 2: *Evaluation result: precision@k, corresponds to the number of problematic dialogues in the top k ranked options.*

<b>Precision@</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>
DBDC3	.700	.800	.800	.825
BiLSTM+MLP	<b>1.00</b>	.975	<b>.953</b>	<b>.937</b>
HierAttn	<b>1.00</b>	<b>.985</b>	.943	<b>.937</b>
<b>Precision@</b>	<b>50</b>	<b>100</b>	<b>150</b>	<b>200</b>
DBDC3	.800	.720	.713	.655
BiLSTM+MLP	.906	.829	.769	.736
HierAttn	<b>.924</b>	<b>.873</b>	<b>.840</b>	<b>.786</b>

## 6. Conclusions

In this paper, we introduced a new log dialogue ranking task and described an entirely crowd-sourcing based data collection pipeline. The initial benchmark results show the utility of the collected corpus in training a ranker with high performance. To facilitate further research on the log dialogue ranking task, we release the collected data and the software at <https://github.com/XinnuoXu/DRank>. Our future work includes exploring unsupervised data augmentation techniques to effectively reduce the cost of data collection.

## 7. References

- [1] J. D. Williams, K. Asadi, and G. Zweig, "Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning," *arXiv preprint arXiv:1702.03274*, 2017.
- [2] A. Bordes, Y.-L. Boureau, and J. Weston, "Learning end-to-end goal-oriented dialog," *arXiv preprint arXiv:1605.07683*, 2016.
- [3] M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella, "Paradise: A framework for evaluating spoken dialogue agents," *arXiv preprint cmp-lg/9704004*, 1997.
- [4] K.-P. Engelbrecht, F. Gódde, F. Hartard, H. Ketabdar, and S. Möller, "Modeling user satisfaction with hidden markov models," in *Proceedings of the SIGDIAL 2009 Conference*, 2009, pp. 170–177.
- [5] M. A. Walker, I. Langkilde-Geary, H. W. Hastie, J. Wright, and A. Gorin, "Automatically training a problematic dialogue predictor for a spoken dialogue system," *Journal of Artificial Intelligence Research*, vol. 16, pp. 293–319, 2002.
- [6] O. Herm, A. Schmitt, and J. Liscombe, "When calls go wrong: How to detect problematic calls based on log-files and emotions?" in *Ninth Annual Conference of the International Speech Communication Association*, 2008.
- [7] W. Kim, "Online call quality monitoring for automating agent-based call centers," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.
- [8] R. Higashinaka, Y. Minami, K. Dohsaka, and T. Meguro, "Modeling user satisfaction transitions in dialogues from overall ratings," in *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2010, pp. 18–27.
- [9] R. Higashinaka, K. Funakoshi, Y. Kobayashi, and M. Inaba, "The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics," in *LREC*, 2016.
- [10] S. Ultes, A. Schmitt, and W. Minker, "Towards quality-adaptive spoken dialogue management," in *NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data*. Association for Computational Linguistics, 2012, pp. 49–52.
- [11] P.-H. Su, M. Gasic, N. Mrksic, L. Rojas-Barahona, S. Ultes, D. Vandyke, T.-H. Wen, and S. Young, "On-line active reward learning for policy optimisation in spoken dialogue systems," *arXiv preprint arXiv:1605.07669*, 2016.
- [12] H. H. Clark, *Using language*. Cambridge university press, 1996.
- [13] D. Bohus and A. I. Rudnicky, "Sorry, i didn't catch that!-an investigation of non-understanding errors and recovery strategies," in *6th SIGdial workshop on discourse and dialogue*, 2005.
- [14] R. Higashinaka, K. Funakoshi, M. Araki, H. Tsukahara, Y. Kobayashi, and M. Mizukami, "Towards taxonomy of errors in chat-oriented dialogue systems," in *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2015, pp. 87–95.
- [15] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.
- [16] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.
- [17] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015, arXiv: 1412.6980. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [18] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. S. Zettlemoyer, "Allennlp: A deep semantic natural language processing platform," 2017.
- [19] R. Higashinaka, K. Funakoshi, M. Inaba, Y. Tsunomori, T. Takahashi, and N. Kaji, "Overview of dialogue breakdown detection challenge 3," *Proceedings of Dialog System Technology Challenge*, vol. 6, 2017.