RSL19BD at DBDC4: Ensemble of Decision Tree-based and LSTM-based Models

Chih-hao Wang and Sosuke Kato and Tetsuya Sakai

Abstract RSL19BD (Waseda University Sakai Laboratory) participated in DBDC4 and submitted five runs to both English and Japanese subtasks. In these runs, we utilise the decision tree-based model and the Long Short-Term Memory-based (LSTM-based) models following RSL17BD and KTH in the Third Dialogue Breakdown Detection Challenge (DBDC3) respectively. Run 1 follows the approach of RSL17BD but utilises RandomForestRegressor instead of ExtraTreesRegressor. In addition, instead of predicting the mean and the variance of the probability distribution of the three breakdown labels for each target system utterance, it predicts the probability of each label directly. Run 2 follows the approach of KTH with some changes in the architecture and utilises Convolutional Neural Network (CNN) to perform text feature extraction. In addition, instead of targeting the single breakdown label and minimising the categorical cross entropy loss for each target system utterance during training process, it targets the probability distribution of the three breakdown labels and minimises its mean squared error. Run 3 performs an ensemble of five LSTM-based models using the same architecture as Run 2. Run 4 performs an ensemble of the models used in Run 1 and Run 2. Run 5 performs an ensemble of the models used in Run 1 and Run 3. Run 5 statistically significantly outperforms all runs in terms of MSE (NB, PB, B) for the English data and outperforms all runs except Run 4 in terms of MSE (NB, PB, B) for the Japanese data (alpha level = 0.05).

Chih-hao Wang
Waseda University, Shinjuku-ku Okubo 3-4-1 Tokyo Japan,
e-mail: haohaowang.oscar@moegi.waseda.jp

Sosuke Kato
Waseda University, Shinjuku-ku Okubo 3-4-1 Tokyo Japan, e-mail: sow@suou.waseda.jp

Tetsuya Sakai
Waseda University, Shinjuku-ku Okubo 3-4-1 Tokyo Japan, e-mail: tetsuyasakai@acm.org

1 Introduction

The task in the Fourth Dialogue Breakdown Detection Challenge (DBDC4) [3] is to build a model that detects whether an utterance from the system causes a breakdown in a dialogue context involving a system and a user. A breakdown is defined as a situation where a user cannot proceed with the conversation. Given a system utterance, the model is required to produce two outputs: 1. A single breakdown label chosen from the three breakdown labels (NB: Not a breakdown, PB: possible breakdown, and B: breakdown) which indicate whether a system utterance causes a breakdown. 2. The probability distribution of the three breakdown labels, which we refer as P(NB), P(PB), and P(B) in this paper. For evaluating the model, the organisers adopted classification-related metrics and distribution-related metrics and put an emphasis on mean squared error (MSE). A complete description of the challenge can be found in [3].

RSL19BD (Waseda University Sakai Laboratory) participated in DBDC4 and submitted five runs to both English and Japanese subtasks. In these runs, we utilise the decision tree-based model and the Long Short-Term Memory-based (LSTM-based) models following RSL17BD [6] and KTH [8] in the Third Dialogue Breakdown Detection Challenge (DBDC3) [4] respectively. Run 1 follows the approach of RSL17BD but utilises RandomForestRegressor [1] [1] instead of ExtraTreesRegressor [2] [2]. In addition, instead of predicting the mean and the variance of the probability distribution of the three breakdown labels for each target system utterance, it predicts the probability of each label directly. Run 2 follows the approach of KTH [8] with some changes in the architecture and utilises Convolutional Neural Network (CNN) to perform text feature extraction. In addition, instead of targeting the single breakdown label and minimising the categorical cross entropy loss for each target system utterance during training process, it targets the probability distribution of the three breakdown labels and minimises its mean squared error. Run 3 performs an ensemble of five LSTM-based models using the same architecture as Run 2. Run 4 performs an ensemble of the models used in Run 1 and Run 2. Run 5 performs an ensemble of the models used in Run 1 and Run 3.

2 Prior Art

At DBDC3, RSL17BD and KTH both submitted models which achieved high performances [4]. This section briefly describes their approaches.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
[2] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html

2.1 RSL17BD at DBDC3

The top-performing model of RSL17BD utilises ExtraTreesRegressor and employed six features shown in Table 1 based on pattern analysis to predict the mean and variance of the probability distribution of the breakdown labels for each target system utterance. The predicted mean and variance are then converted into the predicted probability distribution of the three breakdown labels. The single breakdown label is determined by choosing the label with the highest probability.

Table 1 Features employed by RSL17BD at DBDC3

Feature
turn-index of the target utterance
length of the target utterance (number of characters)
length of the target utterance (number of terms)
keyword flags of the target utterance
term frequency vector similarities among the target system utterance, the immediately preceding user utterance, and the system utterance that immediately precedes that user utterance
word embedding vector similarities among the target system utterance, the immediately preceding user utterance, and the system utterance that immediately precedes that user utterance

2.2 KTH at DBDC3

The top-performing model of KTH utilises Long Short-Term Memory (LSTM) [5]. For the preprocessing of English data, it produces a sequence of 300 dimensional word embedding vectors for each utterance in every dialogue and take the average sum of the sequence to produce the final embedding for a single utterance. The number of turns in each dialogue is fixed to 20 by removing the first system utterance which has no annotations or removing the last user turn. This produces an embedded dialogue of 20 turns, with each turn represented by a single 300 dimensional utterance embedding. The embedded dialogue is then processed by 4 LSTM layers and a Dense layer to produce 4 outputs for each turn. The 4 outputs are P(NB), P(PB), P(B), and P(U), where P(U) refers to the probability of user turn. The reason for adding P(U) is that user turns are included in the embedded dialogue as well and need to be predicted with a label different from NB, PB, and B. The model is trained for 100 epochs using Adadelta [12] as optimiser. During training, it targets the single breakdown label and aims to minimise the categorical cross entropy loss for each target system utterance. For Japanese data, KTH did not submit any runs.

3 Description of DBDC4 Dataset

The development and evaluation dataset given in DBDC4 contains two languages: English and Japanese.

The English data consists of dialogues from a dialogue system named IRIS and six other dialogue systems (anonymised as Bot001 to Bot006) which participated in the conversational intelligence challenge. In this paper, Bot001 to Bot006 are treated as a single system referred as BOT. Each dialogue is composed of 20 or 21 turns of alternating system and user utterances, with 10 system utterances being labeled. The labeled system utterances are evaluated by 15 human annotators, where each annotator labels it with a breakdown label chosen from NB, PB, and B.

The Japanese data consists of two types of dialogues. The first type of dialogues come from three dialogue systems named DCM, DIT, and IRS. Each dialogue is composed of 21 turns of alternating system and user utterances, with 11 system utterances being labeled. The second type of dialogues are located under a folder named `dbd_livecompe_eval`. These dialogues come from five systems (IRS, MMK, MRK, TRF, and ZNK) which participated in a live competition held in Japan. Each dialogue is composed of 31 turns of alternating system and user utterances, with 16 system utterances being labeled. In the development data, all labeled system utterances are evaluated by 30 human annotators. In the evaluation data, labeled system utterances of the first type of dialogues are evaluated by 15 human annotators while the ones of the second type are evaluated by 30 human annotators. A complete description of the dataset can be found in [3].

For the development dataset of each dialogue system, we calculated the average probability distribution of the three breakdown labels across all its utterances. We did not do so for the evaluation dataset since it is unlabeled. Tables 2 to 4 show our calculated results along with other statistical information for each dialogue system.

Table 2 Statistics of DBDC4 English data

System name	No. of dialogues	No. of turns	No. of annotators	NB	PB	B
BOT (dev)	168	20 or 21	15	38.1%	28.7%	33.2%
IRIS (dev)	43	21	15	30.0%	30.3%	39.6%
BOT (eval)	173	20 or 21	15	-	-	-
IRIS (eval)	27	21	15	-	-	-

Table 3 Statistics of DBDC4 Japanese development data from DCM, DIT, and IRS

System name	No. of dialogues	No. of turns	No. of annotators	NB	PB	B
DCM (dev)	350	21	30	42.2%	29.9%	27.9%
DIT (dev)	150	21	30	26.0%	29.6%	44.4%
IRS (dev)	150	21	30	30.5%	25.8%	43.7%
DCM (eval)	50	21	15	-	-	-
DIT (eval)	50	21	15	-	-	-
IRS (eval)	50	21	15	-	-	-

Table 4 Statistics of DBDC4 Japanese development data from the five dialogue systems under `dbd_livecompe_eval`

System name	No. of dialogues	No. of turns	No. of annotators	NB	PB	B
IRS (dev)	13	31	30	32.8%	25.4%	41.7%
MMK (dev)	15	31	30	57.6%	29.4%	13.0%
MRK (dev)	15	31	30	48.5%	35.5%	16.0%
TRF (dev)	14	31	30	69.4%	20.0%	10.6%
ZNK (dev)	16	31	30	47.2%	30.6%	22.2%
IRS (eval)	15	31	30	-	-	-
MMK (eval)	14	31	30	-	-	-
MRK (eval)	14	31	30	-	-	-
TRF (eval)	16	31	30	-	-	-
ZNK (eval)	14	31	30	-	-	-

4 Model Descriptions

4.1 Decision Tree-based model

For the preprocessing of both English and Japanese data, we follow the same approach as RSL17BD [6] at DBDC3 [4]. The model we built employs the same set of features as RSL17BD's model at DBDC3, but utilises RandomForestRegressor instead of ExtraTreesRegressor. In addition, instead of first predicting the mean and the variance of the probability distribution over the three breakdown labels and then deriving the probability of each label, it predicts the probability of each label directly. The probability distribution is then calculated by normalising the probability of the three labels by their sum.

The modifications above have been decided by training and evaluating different model configurations using the English [3] and Japanese data from DBDC3 [4]. The evaluation results are shown in Tables 5 and 6. **DT** means the model utilises decision trees, **EX10** means the model utilises ExtraTreesRegressor with 10 estimators,

[3] When training and evaluating a model using the English data from DBDC3, we used the revised data mentioned in the DBDC3 overview paper [4].

[4] Due to the late release of dataset for DBDC4, we first built our models using the dataset from DBDC3.

and **RF100** means the model utilises RandomForestRegressor with 100 estimators. **AV** means the model predicts the mean and the variance of the probability distribution, while **NBPBB** means the model predicts the probability of each label directly. There are four evaluation metrics. Accuracy denotes the number of correctly predicted breakdown labels divided by the total number of breakdown labels to be predicted (the larger the better); F1 (B) denotes the F1-measure where only the B labels are considered correct (the larger the better); JSD (NB, PB, B) denotes the Jensen-Shannon Divergence between the predicted and correct probability distribution (the smaller the better); MSE (NB, PB, B) denotes the mean squared error between the predicted and correct probability distribution (the smaller the better). The results show that **DT-RF100-NBPBB** outperformed the model submitted by RSL17BD in DBDC3 (**DT-EX10-AV**) in all evaluation metrics in both English and Japanese data. Thus, we chose to utilise the configuration of **DT-RF100-NBPBB** when submitting the model for Run 1.

Table 5 Results of Decision Tree-based model with different configurations on DBDC3 English evaluation data

Model	Accuracy	F1 (B)	JSD (NB, PB, B)	MSE (NB, PB, B)
DT-EX10-AV	0.3430	0.2344	0.0594	0.0357
DT-EX10-NBPBB	0.4065	**0.3696**	0.0498	0.0291
DT-RF10-NBPBB	0.3950	0.3542	0.0486	0.0282
DT-RF100-NBPBB	**0.4095**	0.3548	**0.0466**	**0.0271**

Table 6 Results of Decision Tree-based model with different configurations on DBDC3 Japanese evaluation data

Model	Accuracy	F1 (B)	JSD (NB, PB, B)	MSE (NB, PB, B)
DT-EX10-AV	0.3927	0.3225	0.1297	0.0769
DT-EX10-NBPBB	0.5303	0.6050	0.0920	0.0502
DT-RF10-NBPBB	0.5455	0.6292	0.0875	0.0481
DT-RF100-NBPBB	**0.5630**	**0.6511**	**0.0845**	**0.0460**

4.2 LSTM-based model

Following the approach of KTH [8], we utilise Long Short-Term Memory (LSTM) [5] for this run, but instead of taking the average sum of word embedding vectors for each utterance, we utilise Convolutional Neural Networks (CNN) and Global Max Pooling to perform text feature extractions and produce the final embedded utterance. In addition, instead of targeting the single breakdown label and minimising the categorical cross entropy loss for each target system utterance during training process, our model targets the probability distribution of the three breakdown labels and minimises its mean squared error. We chose Adam [7] as our optimiser and mean squared error as our loss function. Fig. 1 shows the architecture diagram of the model.

For the preprocessing of both English and Japanese data, we first follow the same approach as RSL17BD [6] at DBDC3 [4] and produce a sequence of 300 dimensional word embedding vectors for each utterance in every dialogue. The number of word vectors in each sequence is fixed to v, with v set to 50. This is done by truncating sequences that are longer than v and padding sequences that are shorter than v with zero vectors. The number of turns in a each dialogue is fixed to $2n$ by either removing the first system utterance which has no annotations or removing the last user turn. For English data and DCM, DIT, IRS from Japanese data, the number of turns in each dialogues is fixed to 20 by setting n to 10. For the data of the five dialogue systems under `dbd_livecompe_eval`, the number of turns in each dialogues is fixed to 30 by setting n to 15.

The process above produces a dialogue of $2n$ turns, with each turn represented by a sequence of v word vectors. We apply One-dimensional Convolutional Neural Networks (1D CNN), One-dimensional Global Max Pooling (1D GMax Pooling), and Dropout [11] for each sequence to produce an embedded dialogue. The 1D CNN layer uses 150 filters of size 2 with relu [9] as the activation function. The dropout rate of the Dropout layer is set to 0.4.

The embedded dialogue is then processed by 4 LSTM layers sequentially. Each LSTM layer contains 64 units, with dropout set to 0.1, and recurrent dropout set to 0.1. We used LSTM instead of Bi-LSTM because the usage of turns after the target system utterance is disallowed. The output sequences from the 4 LSTM layers are concatenated to form a ($2n$, 256) dimensional matrix, and processed by a Dense layer with softmax activation and 4 outputs. The 4 outputs represent P(NB), P(PB), P(B), and P(U) respectively. The probability distribution for each target system utterance is calculated by normalising P(NB), P(PB), and P(B) by their sum. The single breakdown label is determined by choosing the label with the highest probability in the distribution.

The modifications above have been decided by training and evaluating different model configurations using the English and Japanese data from DBDC3. The evaluation results are shown in Tables 7 and 8. **LSTM** means the model utilises LSTM, and **LTSM-CNN** means the model utilises LSTM and CNN. **ADAD-CAT** means the model utilises Adadelta as optimizer and categorical cross entropy as loss function, while **ADAM-MSE** means the model utilises Adam as optimizer and mean squared error as loss function.

The results show that for English data, **LSTM-CNN-ADAM-MSE** outperformed **LSTM-ADAD-CAT** and **LSTM-ADAM-MSE** in all evaluation metrics except F1 (B). Although **LSTM-ADAD-CAT** achieved high performance in F1 (B), its performance in mean squared error (MSE (NB, PB, B)) was poor. Since mean squared error is emphasised in this challenge, we decided to discard **LSTM-ADAD-CAT**. For Japanese data, **LSTM-CNN-ADAM-MSE** outperformed **LSTM-ADAM-MSE** in all evaluation metrics. We did not evaluate **LSTM-ADAD-CAT** because it is already discarded after the evaluation of English data. In the end, we chose to utilise the configuration of **LSTM-CNN-ADAM-MSE** when submitting the models for Run 2 and Run 3.

Table 7 Results of LSTM-based model with different configurations on DBDC3 English evaluation data

Model	Epochs	Accuracy	F1 (B)	JSD (NB, PB, B)	MSE (NB, PB, B)
LSTM-ADAD-CAT	100	0.4130	**0.4616**	0.0928	0.0573
LSTM-ADAM-MSE	100	0.3940	0.3714	0.0516	0.0300
LSTM-CNN-ADAM-MSE	50	**0.4620**	0.4268	**0.0474**	**0.0274**

Table 8 Results of LSTM-based model with different configurations on DBDC3 Japanese evaluation data

Model	Epochs	Accuracy	F1 (B)	JSD (NB, PB, B)	MSE (NB, PB, B)
LSTM-ADAM-MSE	100	0.5448	0.6148	0.0885	0.0497
LSTM-CNN-ADAM-MSE	50	**0.5739**	**0.6594**	**0.0826**	**0.0463**

5 Runs

The descriptions of our runs are shown in Table 9.

Table 9 Description of runs for English and Japanese

Run	Description
1	Decision Tree-based model
2	LSTM-based model
3	Ensemble of 5 LSTM-based models
4	Ensemble of Run 1 and Run 2
5	Ensemble of Run 1 and Run 3

5.1 Run 1: Decision Tree-based model

For the English submission, we trained the model with the entire English development data in DBDC4 and made predictions on the 200 evaluation data. For the

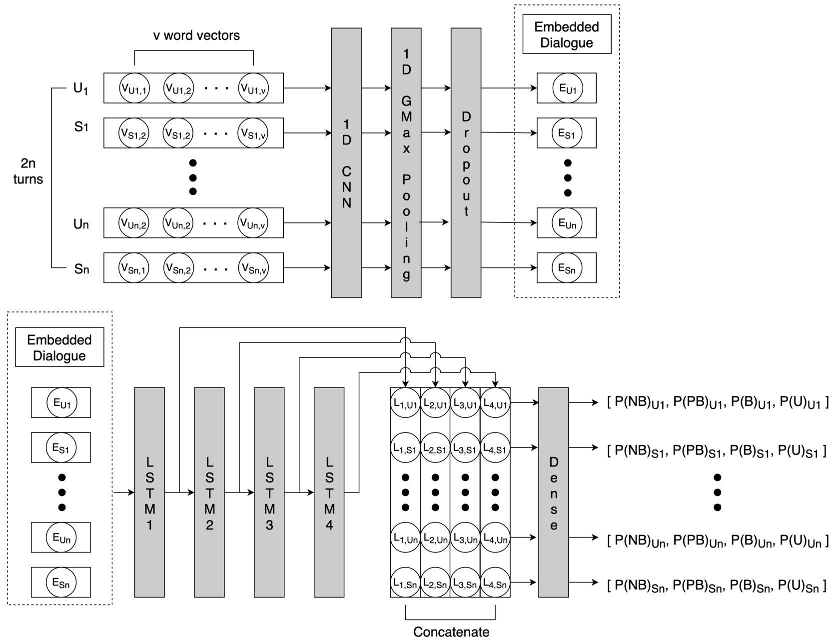

Fig. 1 Architecture diagram of our LSTM-based model

Japanese submission, we built two models by training one with the development data from DCM, DIT, and IRS, and the other with the development data from the five dialogue systems under `dbd_livecompe_eval`. We made predictions on the 150 evaluation data from DCM, DIT, and IRS using the former model and the 73 evaluation data from the five dialogue systems under `dbd_livecompe_eval` using the latter model.

5.2 Run 2: LSTM-based model

For the English submission, we pretrained the model for 30 epochs with the entire English development and evaluation data in DBDC3, fine tuned it by training for 32 epochs with the entire English development data in DBDC4, and made predictions on the 200 evaluation data. For the Japanese submission, we built two models. The first model is trained for 30 epochs with the development data from DCM, DIT, and IRS. The second model is created by loading the pretrained weights from the first model and fine tuned by training for 25 epochs with the development data from the five dialogue systems under `dbd_livecompe_eval`. We made predictions on the 150 evaluation data from DCM, DIT, and IRS using the first model and the 73 evaluation data from the five dialogue systems under `dbd_livecompe_eval` using the second model. Every model is trained using a batch size of 32.

5.3 Run 3: Ensemble of 5 LSTM-based models

In this run, we built our models using the same architecture as Run 2.

For the English submission, we first pretrained a model for 30 epochs with the entire English development and evaluation data in DBDC3. We randomly divide the English development data in DBDC4 into 10 portions and sampled 5 portions from it. We built 5 models by fine tuning the pretrained model, where each model is trained using one of the sampled portions as validation data and the rest of the development data as training data. Each model is saved when the validation loss is minimum and no overfitting occurred. The results of each model on the sampled validation data are shown in Table 10. We made predictions on the 200 evaluation data using each model, and take the mean of the predicted probability distribution for each target system utterance from the 5 models to produce a new probability distribution. We also determine the new single breakdown label by choosing the label with the highest probability in the new probability distribution.

For the Japanese submission of DCM, DIT, and IRS, we randomly divide the development data from DCM, DIT, and IRS into 10 portions and sampled 5 portions from it. We built 5 models by fine tuning the pretrained model, where each model is trained using one of the sampled portions as validation data and the rest of the development data as training data. Each model is saved when the validation loss is minimum and no overfitting occurred. The results of each model on the sampled validation data are shown in Table 11. We made predictions on the 150 evaluation data using each model, and take the mean of the predicted probability distribution for each target system utterance from the 5 models to produce a new probability distribution. We also determine the new single breakdown label by choosing the label with the highest probability in the new probability distribution.

For the Japanese submission of the five dialogue systems under `dbd_livecompe_eval`, we built a pretrained model by preloading the weights from the first model for the Japanese submission in Run 2. We randomly divide the development data into 10 portions and sampled 5 portions from it. We built 5 models by fine tuning the pretrained model, where each model is trained using one of the sampled portions as validation data and the rest of the development data as training data. Each model is saved when the validation loss is minimum and no overfitting occurred. The results of each model on the sampled validation data are shown in Table 12. We made predictions on the 73 evaluation data using each model, and take the mean of the predicted probability distribution for each target system utterance from the 5 models to produce a new probability distribution. We also determine the new single breakdown label by choosing the label with the highest probability in the new probability distribution.

Table 10 Results of each model in Run 3 on the sampled DBDC4 English validation data

Model	Accuracy	F1 (B)	JSD (NB, PB, B)	MSE (NB, PB, B)
1	0.5286	0.5385	0.0649	0.0343
2	0.5190	0.5318	0.0701	0.0370
3	0.5524	0.5660	0.0713	0.0375
4	0.5381	0.6306	0.0706	0.0370
5	0.5810	0.6635	0.0771	0.0401

Table 11 Results of each model in Run 3 on the sampled DBDC4 Japanese validation data from DCM, DIT, and IRS

Model	Accuracy	F1 (B)	JSD (NB, PB, B)	MSE (NB, PB, B)
1	0.5664	0.5782	0.0887	0.0469
2	0.5804	0.6057	0.0914	0.0477
3	0.5944	0.6505	0.0786	0.0429
4	0.5944	0.6402	0.0903	0.0473
5	0.5846	0.6231	0.0961	0.0495

Table 12 Results of each model in Run 3 on the sampled DBDC4 Japanese validation data from the five dialogue systems under `dbd_livecompe_eval`

Model	Accuracy	F1 (B)	JSD (NB, PB, B)	MSE (NB, PB, B)
1	0.6313	0.4583	0.0706	0.0371
2	0.6953	0.4324	0.0591	0.0323
3	0.6484	0.3750	0.0510	0.0277
4	0.6797	0.3529	0.0600	0.0319
5	0.6016	0.0000	0.0678	0.0336

5.4 Run 4: Ensemble of Run 1 and Run 2

For both English and Japanese submissions, we take the mean of the predicted probability distribution for each target system utterance from Run 1 and Run 2 to produce a new probability distribution. We also determine the new single breakdown label by choosing the label with the highest probability in the new probability distribution.

5.5 Run 5: Ensemble of Run 1 and Run 3

For both English and Japanese submissions, we take the mean of the predicted probability distribution for each target system utterance from Run 1 and Run 3 to produce a new probability distribution. We also determine the new single breakdown label by choosing the label with the highest probability in the new probability distribution.

6 Results

Tables 13 and 14 show the official results of our English and Japanese runs, respectively. It can be observed that Run 5 did well on average. For English runs, it outperformed all other runs in all evaluation metrics. For Japanese runs, it outperformed all other runs in JSD (NB, PB, B) and MSE (NB, PB, B).

Tables 15 and 16 show the results of comparing the MSE (NB, PB, B) of Runs 1-5 based on the Randomised Tukey's Honestly Significant Differences (HSD) test. The test is conducted with 10,000 replicates. The p-values are shown alongside with effect sizes (standardised mean differences) [10]. Table 15 shows that Run 5 statistically significantly outperforms all runs in terms of MSE (NB, PB, B) for the English data, Table 16 shows that Run 5 statistically significantly outperforms all runs except Run 4 in terms of MSE (NB, PB, B) for the Japanese data. The p values show that the differences are statistically significant at the alpha level of 0.05.

Table 13 Official results on English data

Model	Accuracy	F1 (B)	JSD (NB, PB, B)	MSE (NB, PB, B)
Run 1	0.4990	0.4411	0.0700	0.0362
Run 2	0.4730	0.4483	0.0725	0.0374
Run 3	0.5200	0.4554	0.0675	0.0346
Run 4	0.5050	0.4650	0.0690	0.0353
Run 5	**0.5255**	**0.4690**	**0.0662**	**0.0336**

Table 14 Official results on Japanese data

Run	Accuracy	F1 (B)	JSD (NB, PB, B)	MSE (NB, PB, B)
Run 1	0.5390	0.4568	0.0975	0.0492
Run 2	0.5412	**0.4613**	0.0989	0.0509
Run 3	**0.5476**	0.4589	0.0967	0.0493
Run 4	0.5412	0.4583	0.0954	0.0480
Run 5	0.5444	0.4603	**0.0947**	**0.0475**

Table 15 P-value based on Randomised Tukey's HSD test/effect sizes for MSE (NB, PB, B) (English)

	Run2	Run3	Run4	Run 5
Run 1	$p = 0.007(-0.110)$	$p < 0.0001(0.139)$	$p = 0.0669(0.080)$	$p < 0.0001(0.227)$
Run 2	-	$p < 0.0001(0.249)$	$p < 0.0001(0.191)$	$p < 0.0001(0.337)$
Run 3	-	-	$p = 0.387(-0.059)$	$p = 0.0415(0.088)$
Run 4	-	-	-	$p < 0.0001(0.146)$

Table 16 P-value based on Randomised Tukey's HSD test/effect sizes for MSE (NB, PB, B) (Japanese)

	Run2	Run3	Run4	Run 5
Run 1	$p = 0.0086(-0.104)$	$p = 1(-0.002)$	$p = 0.0338(0.076)$	$p < 0.0001(0.112)$
Run 2	-	$p = 0.0086(0.102)$	$p < 0.0001(0.181)$	$p < 0.0001(0.216)$
Run 3	-	-	$p = 0.0222(0.079)$	$p < 0.0001(0.114)$
Run 4	-	-	-	$p = 0.6577(0.035)$

7 Discussions

7.1 Naive strategy in creating the training data

In Run 1-3, we used the same strategy in creating the training data from the given development data. For the English submission, we created one training data which consists of the entire English development data in DBDC4 and trained a model with it. The reason for doing so is that we wanted to create sufficient training data, since we only have a total number of 211 dialogues. For the Japanese submission, we created two groups of training data. One consists of the development data from DCM, DIT, and IRS, and the other consists of the development data from the five dialogue systems under `dbd_livecompe_eval`. We trained two models using the two training data respectively. The reason for doing so is that the first group of training data consists of dialogues with 21 turns (fixed to 20 turns in preprocessing) while the second group consists of dialogues with 31 turns (fixed to 30 turns in preprocessing). Since our LSTM-based model only accepts fixed turn lengths, we had to build two models targeting two different turn lengths. We used the same strategy for building our Decision Tree-based model so that the ensemble with the LSTM-based model can be done easily.

Nevertheless, the above strategy is rather naive as it does not consider the overall probability distribution of the three breakdown labels for each dialogue system. As shown in Tables 2 to 4, the average probability distribution of each dialogue system is different from one another. In particular, the system IRS in Table 4 has a significantly higher probability for label B compared to the other four systems. We believe that IRS should not have been combined with the other four systems to create the training data. Furthermore, the model which is trained with this data should not have been used for predicting the evaluation data of IRS under `dbd_livecompe_eval`.

Table 17 shows the detailed official results of MSE (NB, PB, B) for the five dialogue systems under `dbd_livecompe_eval`. It can be observed that due to the naive strategy above, all runs achieved poor performance with regard to IRS. To improve the result, we believe that the development data of IRS should be excluded from other four systems under `dbd_livecompe_eval` and combined with the group of development data from DCM, DIT, and IRS. When predicting the labels for IRS under `dbd_livecompe_eval`, we should utilise the model trained with the data from DCM, DIT, and IRS instead of the one trained with the data from the systems under `dbd_livecompe_eval`. The above strategy requires us to either fix every training data to 30 turns for the LSTM-based model or develop a new model which utilises smaller number of turns for prediction.

Table 17 Official results of MSE (NB, PB, B) for the five dialogue systems under `dbd_livecompe_eval`

	IRS	MMK	MRK	TRF	ZNK
Run 1	0.0662	0.0243	0.0393	0.0282	0.0418
Run 2	0.0606	0.0184	0.0328	0.0230	0.0389
Run 3	0.0602	0.0195	0.0322	0.0231	0.0394
Run 4	0.0606	0.0197	0.0341	0.0236	0.0378
Run 5	0.0608	0.0206	0.0340	0.0239	0.0384

7.2 Ensemble works?

We analysed our runs in terms of MSE (NB, PB, B) (referred as MSE in this section), which is the emphasised evaluation metric in this challenge. From Tables 13 and 14, it can be observed that Run 4 outperformed Run 1 and Run 2, and Run 5 outperformed Run 1 and Run 3 in terms of MSE for both English and Japanese data. To investigate how well the ensemble actually worked for each utterance, we would like to know the number of target system utterances for which the ensemble model outperformed the original models that were ensembled. In this section, we focus on Run 5 which achieved the best performance in terms of MSE and compare its results with Run 1 and Run 3 [5].

Tables 18 and 19 show the number of target system utterances for which each run outperformed the others and

$$V_{1<3,5} = \{v | mse_1(v) < mse_5(v) < mse_3(v), v \in V\}, \tag{1}$$

$$V_{3<1,5} = \{v | mse_3(v) < mse_5(v) < mse_1(v), v \in V\}, \tag{2}$$

$$V_{5<1,3} = \{v | mse_5(v) < mse_1(v) \wedge mse_5(v) < mse_3(v), v \in V\}, \tag{3}$$

where V denotes the set of target system utterances in the evaluation dataset, and $mse_i(v)$ denotes the MSE of Run i given a target system utterance v ($\in V$).

From Tables 18 and 19, it can be observed that the number of target system utterances for which Run 5 outperformed the other runs is relatively small. We plot

[5] When comparing the runs in section 7.2, we remove the first predicted system utterance of every dialogue in Japanese data. This is because the first system utterances in Japanese data are all annotated with the same labels (NB) and are all predicted correctly with MSE = 0.0 by every run.

Table 18 Number of turns for which each Run outperformed the others for the English dataset

a subset of turns V' ($\subset V$)	$\|V'\|$
$V_{1<3,5}$	866
$V_{3<1,5}$	958
$V_{5<1,3}$	176
$\{v \| mse_1(v) < mse_5(v) \wedge mse_3(v) < mse_5(v), v \in V\}$	0

Table 19 Number of turns for which each Run outperformed the others for the Japanese dataset

a subset of turns V' ($\subset V$)	$\|V'\|$
$V_{1<3,5}$	1200
$V_{3<1,5}$	1233
$V_{5<1,3}$	162
$\{v \| mse_1(v) < mse_5(v) \wedge mse_3(v) < mse_5(v), v \in V\}$	0

the relationship of the differences between the MSE of Run 1, Run 3 and Run 5 in Figs. 2 and 3. The x-axis is $mse_1(v) - mse_5(v)$, and the y-axis is $mse_3(v) - mse_5(v)$. The points coloured in blue, orange and green denote the target system utterances that match the condition of $V_{1<3,5}$, $V_{3<1,5}$ and $V_{5<1,5}$ respectively.

By observing Figs. 2 and 3, we think the condition which makes the MSE of Run 5 lower than the ones of Run 1 and Run 3 is that the target system utterance is located in the first quadrant of Figs. 2 and 3.

Tables 20 and 21 show the MSE of Run 1, Run 3 and Run 5 over $V_{1<3,5}$, $V_{3<1,5}$ and $V_{5<1,3}$ respectively.

Table 20 Mean MSE over $V_{1<3,5}$, $V_{3<1,5}$ and $V_{5<1,3}$ for the English dataset

a subset of turns V' ($\subset V$)	Run 1	Run 3	Run 5
$V_{1<3,5}$	0.0270	0.0451	0.0344
$V_{3<1,5}$	0.0481	0.0285	0.0367
$V_{5<1,3}$	0.0159	0.0159	0.0129

Table 21 Mean MSE over $V_{1<3,5}$, $V_{3<1,5}$ and $V_{5<1,3}$ for the Japanese dataset

a subset of turns V' ($\subset V$)	Run 1	Run 3	Run 5
$V_{1<3,5}$	0.0463	0.0721	0.0573
$V_{3<1,5}$	0.0649	0.0399	0.0505
$V_{5<1,3}$	0.0195	0.0194	0.0164

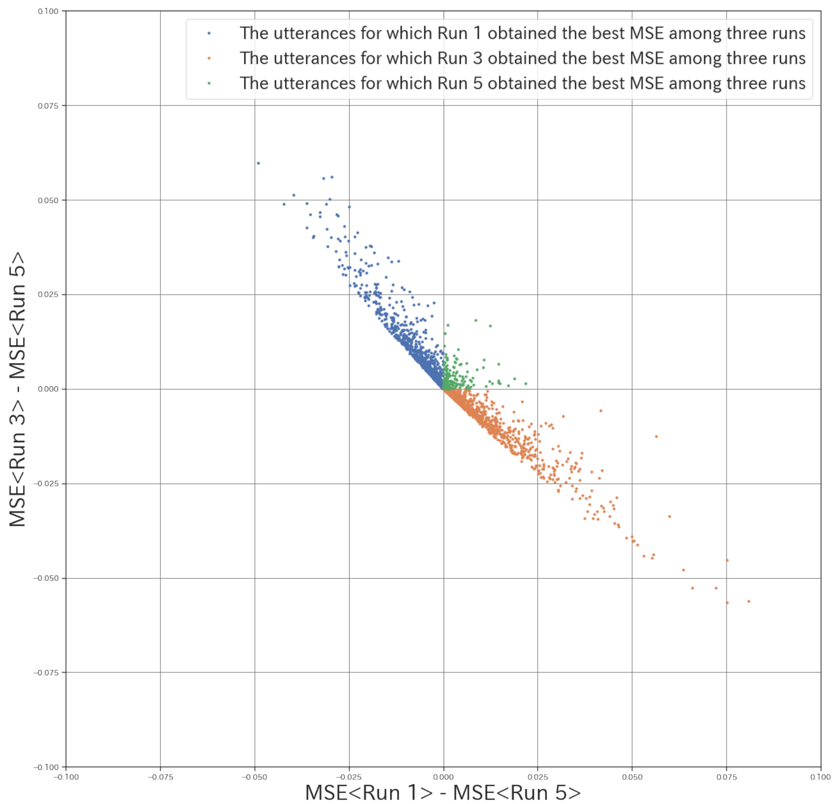

Fig. 2 Relationship of the differences between the MSE of Run 1, Run 3 and Run 5 for the English data

From Tables 20 and 21, it can be observed that when Run 5 outperformed Run 1 and Run 3, the MSEs of Run 1 and Run 3 tend to be low. Similar to Figs. 2 and 3, we plot the relationship of the MSE between Run 1 and Run 3 in Figs. 4 and 5.

By observing Figs. 4 and 5, it appears that the green points are concentrated at the origin of both axes. In addition, Run 5 tends to outperform the two other runs when the MSE of Run 1 and Run 3 are similar. We looked into the system utterances for which the difference between the MSE of Run 1 and Run 3 are high and found out these utterances tend to be labeled with large probability of NB or B compared to other utterances. We plot the relationship of the absolute difference between the MSE of Run 1 and Run 3 and $\max\{p^*(NB), p^*(B)\}$ in Figs. 6 and 7, where $\max\{p^*(NB), p^*(B)\}$ denotes the maximum probability of the labeled probabilities of NB and B. The points coloured in blue are the target system utterances.

From Figs. 6 and 7, it can be observed that the MSE of Run 1 and Run 3 are similar when $\max\{p^*(NB), p^*(B)\}$ is low. This means that the ensemble model tends to perform the best in target system utterances which are not labeled with high probability of NB or B. To further improve our ensemble model, we should either develop

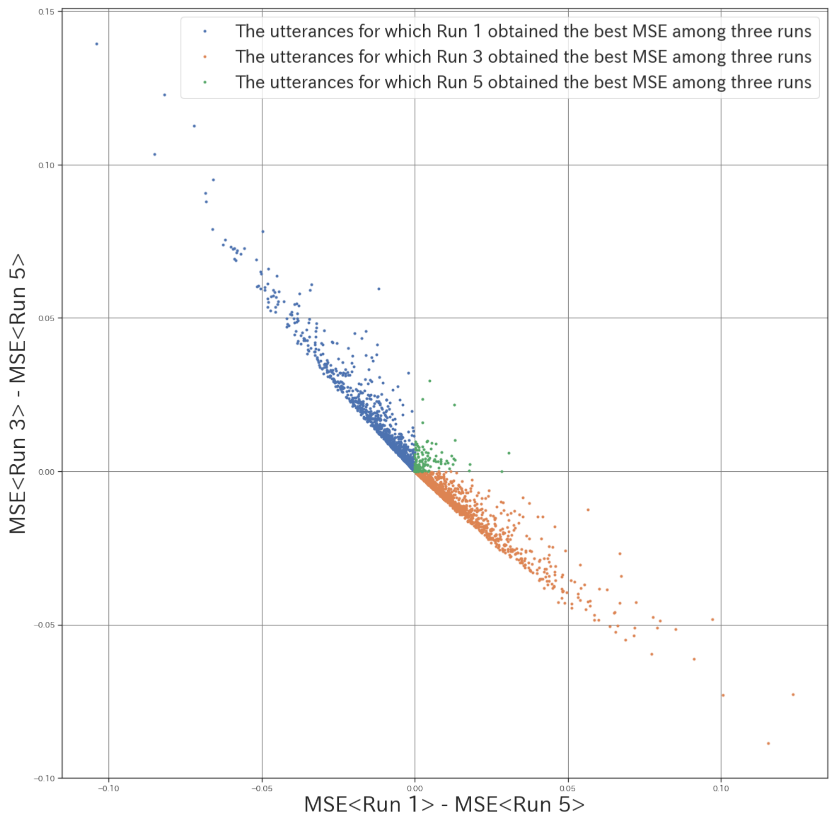

Fig. 3 Relationship of the differences between the MSE of Run 1, Run 3 and Run 5 for the Japanese data

a new ensemble strategy different from simply averaging the two models or include a third model which focus on minimising the MSE in target system utterances that are labeled with high probability of NB or B.

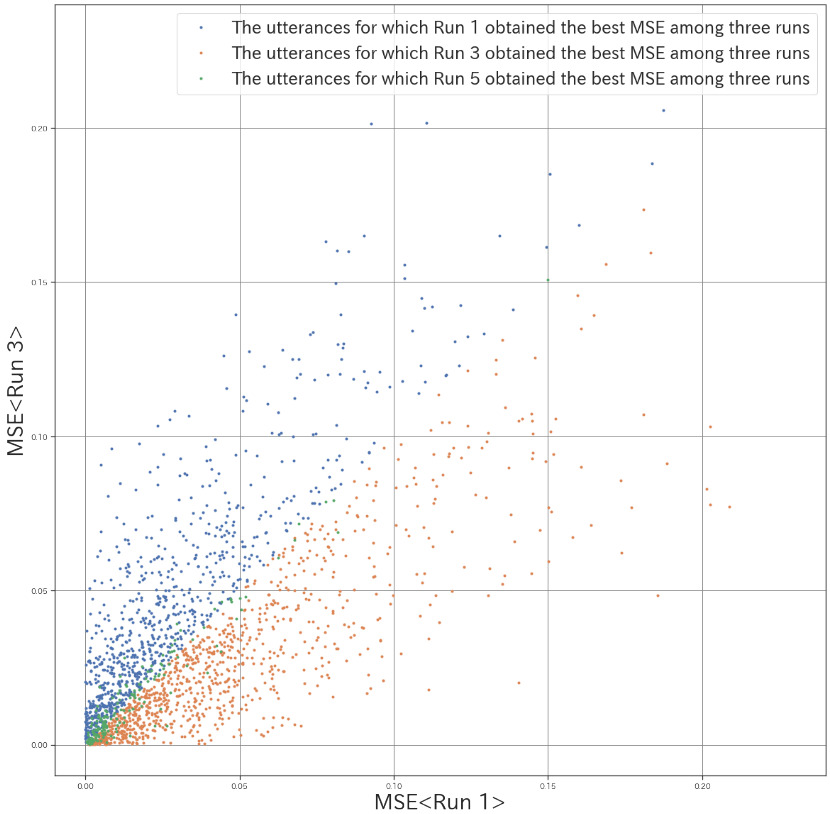

Fig. 4 Relationship of MSEs between Run 1 and Run 3 for the English data

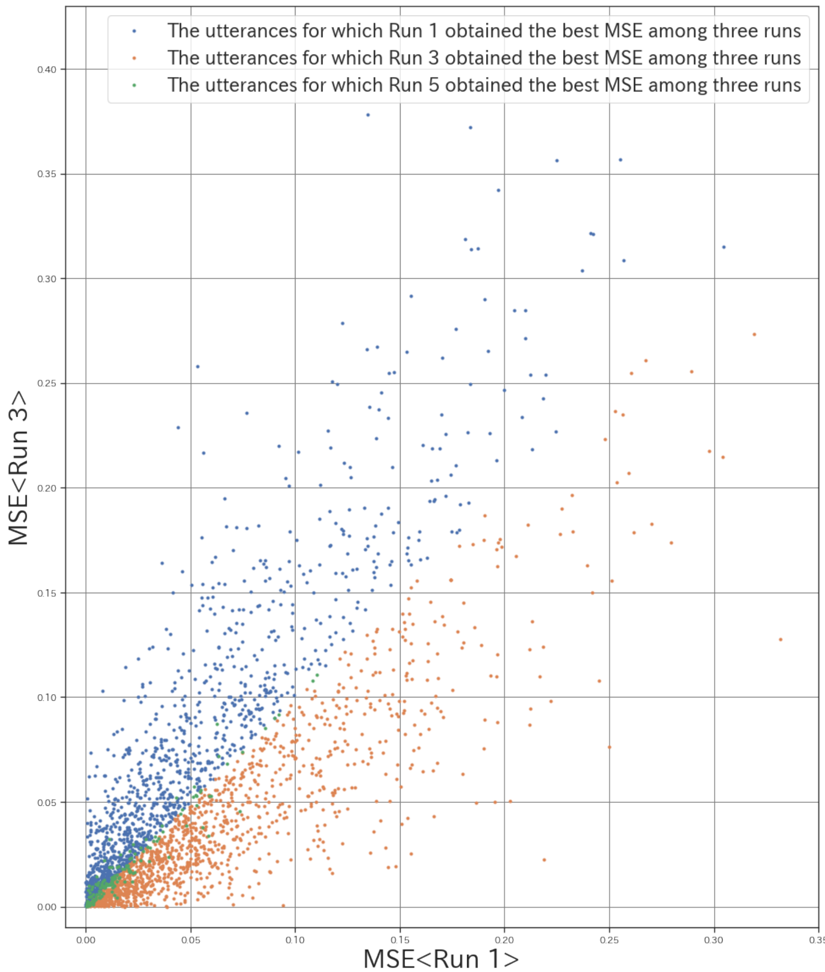

Fig. 5 Relationship of MSEs between Run 1 and Run 3 for the Japanese data

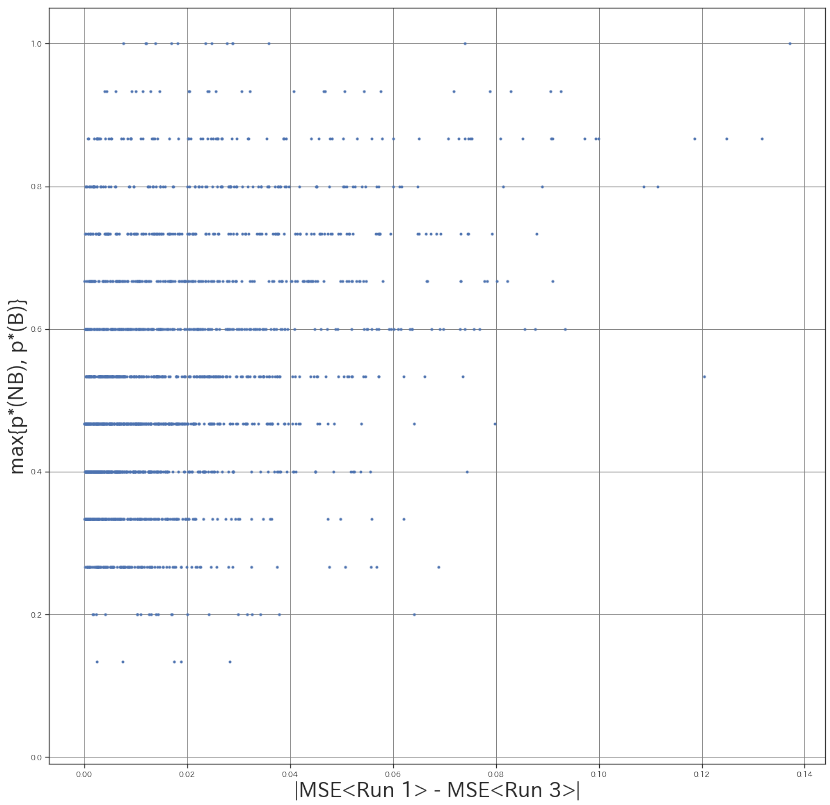

Fig. 6 Relationship of the difference between the MSE of Run 1 and Run 3 and $\max\{p^*(NB), p^*(B)\}$ for the English dataset

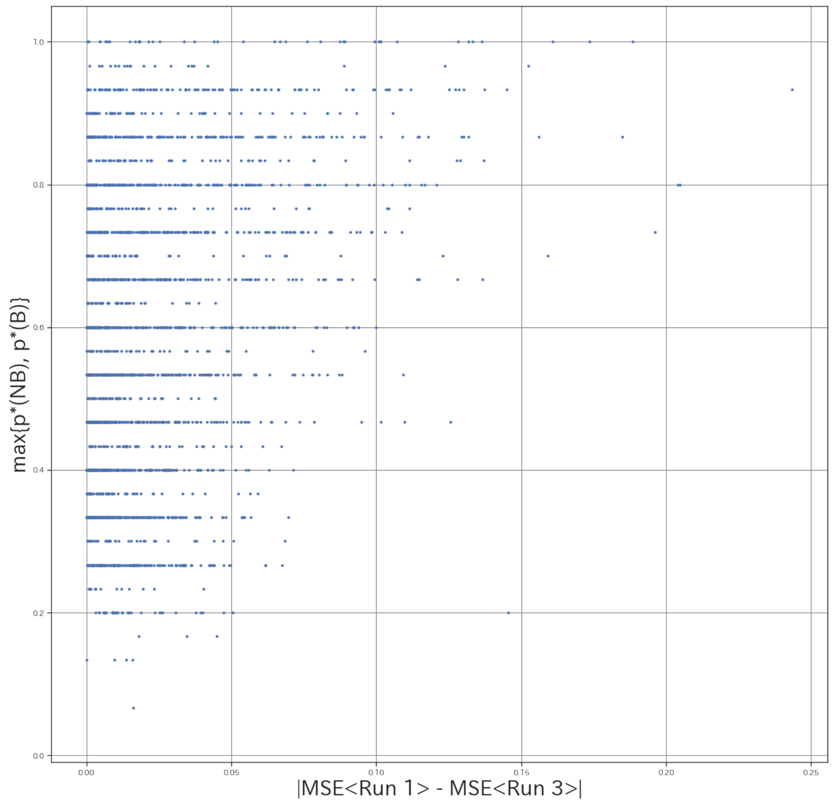

Fig. 7 Relationship of the difference between the MSE of Run 1 and Run 3 and $\max\{p^*(NB), p^*(B)\}$ for the Japanese dataset

8 Conclusions

We submitted five runs to both English and Japanese subtasks of DBDC4. Run 1 utilises a Decision Tree-based model; Run 2 utilises an LSTM-based model; Run3 performs an ensemble of five LSTM-based models using the same architecture as Run 2; Run 4 performs an ensemble of the models used in Run 1 and Run 2; Run 5 performs an ensemble of the models used in Run 1 and Run 3. Run 5 statistically significantly outperforms all runs in terms of MSE (NB, PB, B) for the English data and outperforms all runs except Run 4 in terms of MSE (NB, PB, B) for the Japanese data (alpha level = 0.05).

Our future work includes utilising a better strategy in creating the training data and improving our ensemble model. The better strategy considers the overall probability distribution of the three breakdown labels for each dialogue systems and requires us to either fix every training data to 30 turns for the LSTM-based model or develop a new model which utilises smaller number of turns for prediction. To further improve our ensemble model, we should either develop a new ensemble strategy different from simply averaging the two models or include a third model which focus on minimising the MSE in target system utterances that are labeled with high probability of NB or B.

References

1. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)
2. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning **63**(1), 3–42 (2006). DOI 10.1007/s10994-006-6226-1. URL https://doi.org/10.1007/s10994-006-6226-1
3. Higashinaka, R., D'Haro, L.F., Shawar, B.A., Banchs, R., Funakoshi, K., Inaba, M., Tsunomori, Y., Takahashi, T., ao Sedoc, J.: Overview of the dialogue breakdown detection challenge 4
4. Higashinaka, R., Funakoshi, K., Inaba, M., Tsunomori, Y., Takahashi, T., Kaji, N.: Overview of Dialogue Breakdown Detection Challenge 3. In: Proceedings of Dialog System Technology Challenge 6 (DSTC6) Workshop (2017)
5. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Comput. **9**(8), 1735–1780 (1997)
6. Kato, S., Sakai, T.: RSL17BD at DBDC3: Computing Utterance Similarities based on Term Frequency and Word Embedding Vectors. In: Proceedings of Dialog System Technology Challenge 6 (DSTC6) Workshop (2017)
7. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization (2014)
8. Lopes, J.: How Generic Can Dialogue Breakdown Detection Be? The KTH entry to DBDC3. In: Proceedings of Dialog System Technology Challenge 6 (DSTC6) Workshop (2017)
9. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, pp. 807–814 (2010)
10. Sakai, T.: Laboratory Experiments in Information Retrieval: Sample Sizes, Effect Sizes, and Statistical Power **40**(1) (2018). DOI 10.1007/978-981-13-1199-4. URL https://www.springer.com/jp/book/9789811311987

11. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
12. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. CoRR **abs/1212.5701** (2012). URL http://arxiv.org/abs/1212.5701