A Co-Attentive Cross-Lingual Neural Model for Dialogue Breakdown Detection

Qian Lin[†], **Souvik Kundu**[‡,*] and **Hwee Tou Ng**[†]
[†]Department of Computer Science, National University of Singapore
[‡]Microsoft Dynamics 365 AI
qlin@u.nus.edu, sokun@microsoft.com, nght@comp.nus.edu.sg

Abstract

Ensuring smooth communication is essential in a chat-oriented dialogue system, so that a user can obtain meaningful responses through interactions with the system. Most prior work on dialogue research does not focus on preventing dialogue breakdown. One of the major challenges is that a dialogue system may generate an undesired utterance leading to a dialogue breakdown, which degrades the overall interaction quality. Hence, it is crucial for a machine to detect dialogue breakdowns in an ongoing conversation. In this paper, we propose a novel dialogue breakdown detection model that jointly incorporates a pretrained cross-lingual language model and a co-attention network. Our proposed model leverages effective word embeddings trained on one hundred different languages to generate contextualized representations. Co-attention aims to capture the interaction between the latest utterance and the conversation history, and thereby determines whether the latest utterance causes a dialogue breakdown. Experimental results show that our proposed model outperforms all previous approaches on all evaluation metrics in both the Japanese and English tracks in Dialogue Breakdown Detection Challenge 4 (DBDC4 at IWSDS2019).

1 Introduction

In today's world, we often use interactive virtual conversational agents in our daily lives, such as Siri, Google Assistant, Alexa, etc. Such chat-oriented dialogue systems assist users in their daily lives through interaction in the form of conversations. Through information obtained via a conversation, a system can understand a user's need and generate the corresponding response, such as reporting weather forecasts, booking tickets, or entertaining the user by chit-chatting. To facilitate hassle-free communication, the system needs to generate proper and rational responses without any ambiguity. Furthermore, while generating a response, the system should consider not only the last turn in the conversation but also all previous utterances so that the response generated is coherent with the entire conversation history. The user experience degrades when a dialogue breakdown occurs, where the system generates an inappropriate response which breaks the conversation flow. Dialogue breakdown detection is one of the major

User utterances	System utterances	Breakdown?
U_1: What do you like to do?	S_1: I like to eat out and eat.	NO
U_2: I like to eat too. Hamburgers and french fries are my favorite.	S_2: I love fries. I love cheese.	NO
U_3: Where do you work?	S_3: I work for a company.	NO
U_4: As a what?	S_4: I have a lot of friends.	YES

Table 1: An example dialogue where the last system-generated response causes a dialogue breakdown. U_i and S_j denote the i-th user utterance and the j-th system utterance in the dialogue, respectively.

[*] This work was done when Souvik Kundu was at the National University of Singapore.

Proceedings of the 28th International Conference on Computational Linguistics, pages 4201–4210
Barcelona, Spain (Online), December 8-13, 2020

challenges faced by current chat-oriented dialogue systems, but it has still not been carefully studied by the research community.

The dialogue breakdown detection task is designed to test a system's capability of identifying the undesired utterance causing a dialogue breakdown, which is expected to further help us to build more fluent dialogue systems. The dialogue breakdown detection task requires a participating system to determine whether an utterance generated by a system causes a dialogue breakdown (Higashinaka et al., 2016). Table 1 shows an example of a dialogue breakdown in an ongoing conversation. In this example, the last system response causes a dialogue breakdown, since it does not answer the question "As a what?" in the last user utterance but instead gives a completely irrelevant response "I have a lot of friends."

Dialogue Breakdown Detection Challenge 4 (DBDC4)[1] is a shared task dedicated to dialogue breakdown detection. In this paper, we use the dataset released in DBDC4 for our experiments. Since whether a system utterance causes a dialogue breakdown is somewhat subjective, the task is modeled as a classification task with three classes: Breakdown (B), Possible Breakdown (PB), and Not a Breakdown (NB). Every single instance in the dataset is annotated by multiple annotators. Each instance is assigned a gold-standard (majority) class, and a probability distribution based on all annotators' predictions. The task has two tracks involving two different languages: Japanese and English. The evaluation metrics consist of both classification-related metrics and distribution-related metrics. We will introduce them in detail in Section 4.3.

Prior work has exploited feature-engineered machine learning approaches like random forests, neural network architectures such as LSTM (Hendriksen et al., 2019; Shin et al., 2019; Wang et al., 2019), and the monolingual pretrained language model BERT (Devlin et al., 2019; Sugiyama, 2019). Most prior work treats the dialogue history and the last system response in the same manner. In feature-based approaches (Wang et al., 2019), the features are extracted from the concatenation of both the dialogue history and the last system response, while in other models using pretrained word embeddings (Hendriksen et al., 2019; Shin et al., 2019; Sugiyama, 2019), the interaction between the dialogue history and the last system response is also not explicitly captured.

Recently, cross-lingual language models such as XLM (Conneau and Lample, 2019) and XLM-R (Conneau et al., 2020) demonstrate strong performance on many cross-lingual natural language processing (NLP) tasks (Wang et al., 2018; Conneau et al., 2018). They also outperform pretrained monolingual language models on tasks with low-resource languages. By utilizing shared word embeddings over different languages and multilingual parallel texts, cross-lingual language models encode input texts into one single representation space shared by all languages. This removes the cost of language-specific training. In this paper, we utilize pretrained cross-lingual language models to benefit from the multilingual training data. To the best of our knowledge, ours is the first work to incorporate pretrained cross-lingual language models in dialogue breakdown detection.

Typically, pretrained language models are not trained in a task-specific setting. When we utilize them for a particular task, it might not perform well if the training data is not too large. To better capture the interaction between the previous dialogue history and the last utterance, we propose to integrate a co-attention network with the cross-lingual language model. Experimental results show that our co-attention network significantly improves the performance of our model on the DBDC4 dataset. The source code and trained models of this paper are available at `https://github.com/nusnlp/CXM`.

2 Task Overview

A dialogue history $\mathcal{H}$ consists of a sequence of alternating user and system utterances. The target utterance $\mathcal{T}$ for dialogue breakdown detection is the succeeding system utterance. Each instance $(\mathcal{H}, \mathcal{T})$ is assigned one of three candidate classes: Breakdown (B), Possible Breakdown (PB), and Not a Breakdown (NB). The output of a model includes two components: a predicted class from one of the three candidates $\{$B, PB, NB$\}$, and a probability distribution over the three classes. DBDC4 includes two tracks in two different languages (Japanese and English) with the same task setting.

[1] `https://sites.google.com/site/dialoguebreakdowndetection4/`

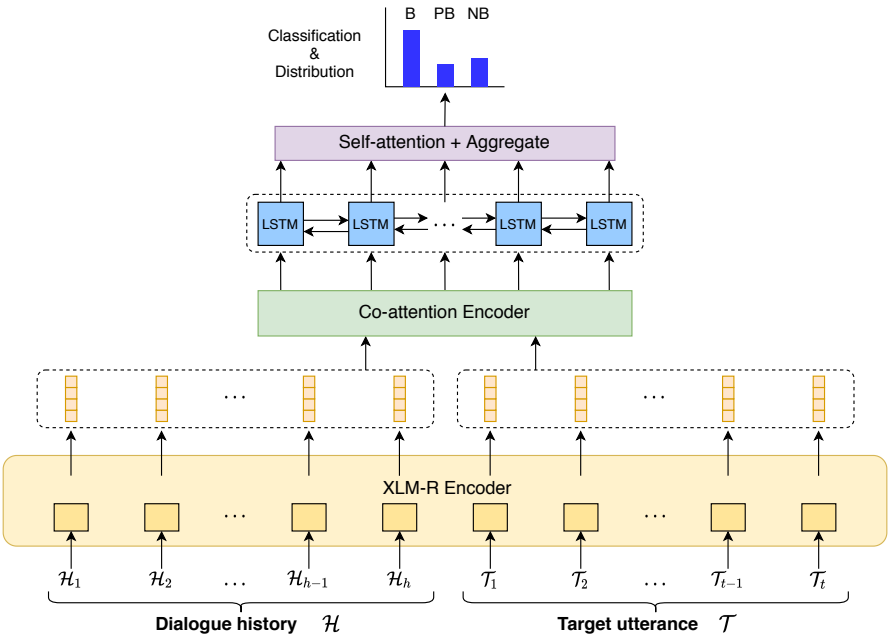

Figure 1: The architecture of our proposed model for DBDC4. $\mathcal{H}_i$ denotes the i-th token in the tokenized dialogue history and $\mathcal{T}_j$ denotes the j-th token in the tokenized target utterance.

3 Model Description

We propose a **C**o-attentive **Cross**-lingual Neural **M**odel (**CXM**), which is based on a pretrained cross-lingual language model and a co-attention network to tackle the task of DBDC4. We give a detailed description of CXM in this section. We present the overall architecture of the proposed model in Figure 1.

3.1 Pretrained Cross-lingual Embeddings

For the embedding layer, we adopt a state-of-the-art cross-lingual pretrained language model named XLM-R (Conneau et al., 2020), which is pretrained on large-scale multilingual corpora. Compared with other cross-lingual language models mBERT (Devlin et al., 2019) and XLM (Conneau and Lample, 2019), the data used to pretrain XLM-R is enlarged by orders of magnitude, especially for low-resource languages. The CC-100 corpus used for pretraining XLM-R includes significantly larger size of both Japanese and English texts than the Wiki-100 corpus on which mBERT and XLM are pretrained (Conneau et al., 2020). In this paper, we choose the largest model of XLM-R with hidden dimension size $d = 1024$.

Assume that the dialogue history $\mathcal{H}$ consists of user-system utterances with h tokens after tokenization, and the target system utterance $\mathcal{T}$ consists of t tokens. We first concatenate these two parts into a combined input representation to XLM-R, and then we obtain the last layer output, denoted as $\mathbf{G} \in \mathbb{R}^{(h+t)\times d}$:

$$\mathbf{G} = \mathrm{emb}([\mathcal{H}; \mathcal{T}]) = [\mathbf{H}; \mathbf{T}] \ , \tag{1}$$

where d is the hidden dimension size of XLM-R, $\mathbf{H} \in \mathbb{R}^{h\times d}$ is the last layer output corresponding to the words in the dialogue history, and $\mathbf{T} \in \mathbb{R}^{t\times d}$ is the last layer output corresponding to the words in the target utterance. With XLM-R embeddings, we utilize data from other languages to enrich the training dataset during training, so as to transfer the knowledge learned from other languages to the target language.

3.2 Co-Attentive Encoding

To further capture the interaction, we propose to utilize a co-attentive encoder to compute a combined representation of the dialogue history and the target utterance. The co-attention network (Lu et al., 2016)

was initially proposed for visual question answering (VQA). In VQA, it was used to jointly reason over image and question attention. Xiong et al. (2017) adapted the co-attention network for machine reading comprehension (e.g., SQuAD). They used a co-attention network to capture the interaction between the question and passage for answer span extraction. While Xiong et al. (2017) used the co-attentive encoder for single-turn machine reading comprehension, we utilize the co-attentive encoder in a more complex dialogue breakdown detection task in a multi-turn conversation setting. The proposed approach of utilizing the co-attentive encoder is described as follows.

To capture the interaction between the dialogue history $\mathcal{H}$ and the target utterance $\mathcal{T}$, we use a co-attentive encoder to compute a combined representation of $\mathcal{H}$ and $\mathcal{T}$. We first calculate the token-wise contextual similarity $\mathbf{A} \in \mathbb{R}^{t \times h}$ between the dialogue history and the target utterance.

$$\mathbf{A} = \mathbf{T}\,\mathbf{H}^{\top} \tag{2}$$

where $\mathbf{A}_{i,j}$ calculates the similarity of the i-th token in the target utterance (i.e., the i-th row of $\mathbf{T}$) and the j-th token in previous dialogue utterances (i.e., the j-th row of $\mathbf{H}$).

We apply a row-wise softmax function for normalization to produce the attention scores over the dialogue history for each word in the target utterance, resulting in $\tilde{\mathbf{A}}$. Next, we compute a summary or weighted representation of the dialogue history corresponding to each word of the target utterance:

$$\mathbf{T}_x \;=\; \tilde{\mathbf{A}}\,\mathbf{H} \;\in\; \mathbb{R}^{t \times d} \tag{3}$$

Next, the initial target utterance encoding vectors in $\mathbf{T}$ (i.e., each row of $\mathbf{T}$), and the vectors in $\mathbf{T}_x$ are concatenated, and passed through a bidirectional LSTM, which results in $\tilde{\mathbf{T}} \in \mathbb{R}^{t \times d}$. $\tilde{\mathbf{T}}$ essentially captures the interaction between the dialogue history and the target utterance by jointly encoding them.

We use a self-attention layer on top of $\tilde{\mathbf{T}}$, which can effectively aggregate evidence from the joint encoding vectors to infer the output. First, we compute the self-attention matrix:

$$\mathbf{A}_s \;=\; \tilde{\mathbf{T}}\,\mathbf{W}\,\tilde{\mathbf{T}}^{\top} \in \mathbb{R}^{t \times t} \;\;, \tag{4}$$

where $\mathbf{W} \;\in\; \mathbb{R}^{d \times d}$ is a trainable bi-linear matrix. Next, we apply a row-wise softmax function for normalization, resulting in $\tilde{\mathbf{A}}_s$. Now, the self-attentive encoding vectors can be aggregated as:

$$\mathbf{T}_s \;=\; \tilde{\mathbf{A}}_s\tilde{\mathbf{T}} \in \mathbb{R}^{t \times d} \tag{5}$$

Then, we concatenate the joint encoding vectors in $\tilde{\mathbf{T}}$ with the self-attentive encoding vectors in $\mathbf{T}_s$, followed by a feed-forward layer, which results in $\mathbf{Y} \in \mathbb{R}^{t \times d}$.

We aggregate the vectors in $\mathbf{Y}$ for the output layer. If the ith row of $\mathbf{Y}$ is represented as $\mathbf{y}_i \in \mathbb{R}^d$, the aggregated vector $\mathbf{v} \in \mathbb{R}^d$ can be written as:

$$\alpha_i \;\propto\; \exp(\mathbf{y}_i\,\mathbf{w}^{\top})\,; \quad \mathbf{v} \;=\; \boldsymbol{\alpha}\,\mathbf{Y} \tag{6}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^t$ and $\mathbf{w} \in \mathbb{R}^d$ is a trainable vector.

Output Layer: In the output layer, we use a simple feed-forward layer on top of $\mathbf{v}$. The number of output units is three for the three different classes. For training, we minimize the mean squared error (MSE) loss summing over all the training instances.

4 Experiments

4.1 Datasets

In our experiments, we focus on the official annotated dataset of DBDC4 including both Japanese and English versions. Following the practice of the DBDC4 participating systems (Hendriksen et al., 2019; Sugiyama, 2019; Wang et al., 2019; Shin et al., 2019), we also utilize the official annotated data released in previous Dialogue Breakdown Detection Challenges including DBDC1, DBDC2, and DBDC3.

We present the dataset statistics in Table 2. We adopt the split of the training set and development set as described in (Sugiyama, 2019). Each dialogue session consists of multiple turns of user-system

Japanese	Train					Dev	Test
Dataset	DBDC1 -JP-dev	DBDC1 -JP-eval	DBDC2 -JP-dev	DBDC2 -JP-eval	DBDC4 -JP-dev	DBDC3 -JP-eval	DBDC4 -JP-eval
# sessions	20	80	150	150	73	150	223
# instances	220	880	1650	1650	1168	1650	2818

English	Train		Dev	Test
Dataset	DBDC3 -EN-dev	DBDC4 -EN-dev	DBDC3 -EN-eval	DBDC4 -EN-eval
# sessions	415	211	200	200
# instances	4150	2110	2000	2000

Table 2: Statistics of the datasets. The upper table shows the Japanese datasets and the lower table shows the English datasets.

interactions. The gold-standard distributions over the three labels for the evaluation on distribution-related metrics are calculated based on labeling decisions from all the annotators. The gold-standard labels for the evaluation on classification-related metrics are decided by majority voting. In the Japanese track, the official test set is DBDC4-JP-eval which consists of 2818 instances. In the English track, the official test set is DBDC4-EN-eval with 2000 instances.

Previous models (Hendriksen et al., 2019; Sugiyama, 2019; Wang et al., 2019; Shin et al., 2019) tested on DBDC4 used only the official datasets for training in a monolingual setting, i.e., the model only incorporated English training data in the English track and Japanese training data in the Japanese track. For training and validation, we use data released in DBDC1, DBDC2, DBDC3, and DBDC4. We evaluate our model on the DBDC4 test set.

4.2 Training

We prepare the training data for our proposed CXM model in two ways. We denote the model trained on single language data as **CXM-S** and on two languages as **CXM-D**. Specifically, in the Japanese track, CXM-S utilizes Japanese training set during training, including DBDC1-JP-dev, DBDC1-JP-eval, DBDC2-JP-dev, DBDC2-JP-eval, and DBDC4-JP-dev. Based on CXM-S, CXM-D adds additional English data during training, including DBDC3-EN-dev, DBDC3-EN-eval, and DBDC4-EN-dev. The development set for both CXM-S and CXM-D is DBDC3-JP-eval.

Similarly in the English track, the training set for CXM-S includes DBDC3-EN-dev and DBDC4-EN-dev, while CXM-D utilizes additional Japanese training data consisting of DBDC1-JP-dev, DBDC1-JP-eval, DBDC2-JP-dev, DBDC2-JP-eval, DBDC3-JP-eval, and DBDC4-JP-dev. The development set is DBDC3-EN-eval.

The parameters in the XLM-R embedding layer are pretrained on corpora of one hundred languages (Conneau et al., 2020) and updated during training. We use mean squared error for the loss function during training.

4.3 Evaluation Metrics

The official evaluation metrics of DBDC4 include classification-based and distribution-based metrics.

Classification: The classification-based metrics consist of accuracy and F1 scores.

- Accuracy: the number of correctly classified instances divided by the total number of instances.
- F1 (B): F1 score corresponding to the classification label B.
- F1 (PB+B): F1 score corresponding to the classification labels PB and B grouped together.

For classification-related metrics, a higher score indicates better performance.

Distribution: The distribution-based metrics utilize Jensen-Shannon divergence (JSD) and Mean Squared Error (MSE).

- JSD (NB, PB, B): Jensen-Shannon divergence calculated over the predicted distribution of all three labels.
- JSD (NB, PB+B): JSD calculated over the predicted distribution of two labels (PB and B are combined and considered as one single label).
- JSD (NB+PB, B): Same as JSD (NB, PB+B) except that NB and PB are combined.
- MSE (NB, PB, B), MSE (NB, PB+B), and MSE (NB+PB, B): Similar to the above metrics except using mean squared error instead of Jensen-Shannon divergence.

Distribution-based metrics measure the difference between the predicted distribution and the gold-standard distribution, so a lower score indicates better performance.

4.4 Participating DBDC4 Models

We compare our proposed CXM model with all previous participating models in DBDC4. We give brief descriptions of these models as follows.
NTTCS19 (Sugiyama, 2019): This model uses a BERT classifier on inputs consisting of the dialogue history, target utterance, and other textual features. The model participated in both Japanese track and English track.
RSL19BD (Wang et al., 2019): This model includes a feature-based random forest regression model, a feature-based LSTM model, and an ensemble of component models. The model participated in both Japanese track and English track.
LIIR (Hendriksen et al., 2019): This is an LSTM model with GloVe embeddings, with inputs consisting of the dialogue history and the target system utterance. The model participated in the English track only.
BitTalk (Shin et al., 2019): This model is a bidirectional LSTM with self-attention on inputs consisting of the dialogue history and the target system utterance. The model participated in the English track only.
Baseline (Higashinaka et al., 2019): This is a conditional random field (CRF) baseline model with textual features extracted from the dialogue history and the target utterance. The model was created by the DBDC4 organizers and it participated in both Japanese and English tracks.

5 Results

In this section, we present the comparison results and an ablation study to better understand our model.

5.1 Model Comparison

We present the comparison results on DBDC4 Japanese track and English track in Table 3 and Table 4, respectively. We denote **NTTCS19** models as **NTT**, **RSL19BD** models as **RSL**, and the organizer's **Baseline** as **BL**. The number following the model name is the index of a run. The scores of previous participating models are retrieved from (Higashinaka et al., 2019). The reported scores of our models are calculated by the official evaluation script provided by the organizers. The results show that our best-performing model CXM-D outperforms all previous models significantly on every evaluation metric in both Japanese and English tracks.

In the Japanese track, when evaluated on classification-based metrics, CXM-D outperforms the best previous models by 13.13%, 13.25%, and 5.57% on accuracy, F1 (B), and F1 (PB+B), respectively. CXM-D improves the previous best models by 34.7%, 32.0%, and 45.6% when evaluated on three JSD-based metrics. When evaluated on MSE-based metrics, the improvements are 36.9%, 39.7%, and 46.0%. For each of the JSD and MSE metrics, the percentage of improvement is obtained by $100 - (\text{ours}/\text{previous\_best}) \times 100$.

In the English track, CXM-D outperforms the previous best models by 7.60%, 11.35%, and 0.92% on accuracy, F1 (B), and F1 (PB+B), respectively. CXM-D improves the previous best models by 14.8%, 13.6%, and 22.4% on the three JSD-based metrics. On the three MSE-based metrics, the improvements are 10.7%, 14.4%, and 14.6%.

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)
BL	0.5330	0.4367	0.6592	0.3839	0.2628	0.2342	0.1997	0.2293	0.2085
NTT-1	0.5841	0.5387	0.7091	0.0953	0.0620	0.0612	0.0463	0.0635	0.0507
NTT-2	0.5724	0.5255	0.7254	0.1014	0.0642	0.0674	0.0504	0.0670	0.0581
NTT-3	0.4808	0.4605	0.7011	0.1259	0.0840	0.0871	0.0653	0.0915	0.0779
NTT-4	0.4446	0.4512	0.7026	0.1361	0.0937	0.0933	0.0715	0.1028	0.0848
RSL-1	0.5390	0.4568	0.6560	0.0975	0.0627	0.0647	0.0492	0.0671	0.0568
RSL-2	0.5412	0.4613	0.7014	0.0989	0.0623	0.0684	0.0509	0.0674	0.0622
RSL-3	0.5476	0.4589	0.6811	0.0967	0.0615	0.0659	0.0493	0.0662	0.0591
RSL-4	0.5412	0.4583	0.6782	0.0954	0.0602	0.0646	0.0480	0.0643	0.0568
RSL-5	0.5444	0.4603	0.6667	0.0947	0.0601	0.0636	0.0475	0.0640	0.0556
CXM-S	0.6831	0.6508	0.7710	0.0734	0.0489	0.0411	0.0376	0.0489	0.0371
CXM-D	**0.7154**	**0.6712**	**0.7811**	**0.0618**	**0.0409**	**0.0333**	**0.0292**	**0.0383**	**0.0274**

Table 3: Experimental results on the DBDC4 Japanese track.

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)
BL	0.4635	0.3421	0.5803	0.4381	0.3176	0.2670	0.2237	0.2788	0.2295
BitTalk	0.4355	0.3901	0.6492	0.0992	0.0706	0.0570	0.0506	0.0734	0.0513
LIIR	0.5335	0.0981	0.0984	0.4245	0.4193	0.4186	0.1323	0.2641	0.2853
NTT-1	0.4880	0.4641	0.7664	0.0733	0.0389	0.0504	0.0378	0.0444	0.0465
NTT-2	0.5320	0.4482	0.6369	0.0752	0.0391	0.0481	0.0377	0.0432	0.0429
NTT-3	0.5345	0.4547	0.6724	0.0709	0.0417	0.0449	0.0361	0.0472	0.0399
NTT-4	0.5560	0.4403	0.6079	0.0693	0.0407	0.0433	0.0351	0.0455	0.0384
RSL-1	0.4990	0.4411	0.6740	0.0700	0.0420	0.0438	0.0362	0.0480	0.0398
RSL-2	0.4730	0.4483	0.7276	0.0725	0.0439	0.0462	0.0374	0.0506	0.0414
RSL-3	0.5200	0.4554	0.6961	0.0675	0.0401	0.0424	0.0346	0.0455	0.0381
RSL-4	0.5050	0.4650	0.7174	0.0690	0.0412	0.0438	0.0353	0.0469	0.0389
RSL-5	0.5255	0.4690	0.6947	0.0662	0.0389	0.0416	0.0336	0.0439	0.0369
CXM-S	0.6205	0.5303	0.6471	0.0586	0.0351	0.0333	0.0312	0.0396	0.0318
CXM-D	**0.6320**	**0.5825**	**0.7756**	**0.0564**	**0.0336**	**0.0323**	**0.0300**	**0.0370**	**0.0315**

Table 4: Experimental results on the DBDC4 English track.

More importantly, CXM-D gives the best scores on all metrics in both DBDC4 Japanese track and English track. It is the first model to achieve dominance on this task. Additionally, we show that when trained on the same monolingual training data as previous models, CXM-S still achieves excellent performance. CXM-S outperforms all previous models on all metrics in the Japanese track, and on all metrics except F1 (PB+B) in the English track.

5.2 Ablation Study

We conduct an ablation study on our proposed model CXM-D on the development set, in order to better understand the effectiveness of our proposed model. We present the results of our ablation study in Table 5. First, we experiment with removing the co-attention component. In this case, we use the output from the first position (`[CLS]`) of the XLM-R last layer output as the contextualized representation. We then use a feed-forward layer followed by a softmax function to output the probability distribution over three labels. Next, we experiment with training the model with single-language data, similar to prior work (Sugiyama, 2019).

By removing the co-attention component, the accuracy drops by 1.77% and 1.65% on the Japanese track and English track, respectively. This indicates that co-attention does better capture the interaction between the dialogue history and the target system utterance. If we use only single-language data during training, the accuracy also drops by 2.01% and 2.85% on the Japanese track and English track, respectively. This indicates that transfer learning from other languages to the target language improves the performance in this task. We observe a further decrease in accuracy on both tracks if we do not use both the co-attention network and the dual-language data.

We analyze 100 randomly selected examples from the development set where CXM-D predicts correctly but the model without co-attention fails. Two examples are given in Table 6. It is evident that

Model	Accuracy (Japanese)	Accuracy (English)
Full model (CXM-D)	**68.74**	**54.70**
– Co-attention	66.97	53.05
– Other lang	66.73	51.85
– Co-attention, other lang	65.76	49.40

Table 5: Ablation study of our proposed model on the development set.

Error type	Dialogue history	Target utterance	Prediction – Co-attention	Gold
Similar topic	... S: Let's talk about movies. U: I love movies.	S: I like pop music.	NB	B
Irrelevant response	... U: Did you vote for President last year? S: I did. U: I did too. Who did you vote for?	S: Thanks. Glad to hear that.	NB	B

Table 6: Examples of different error types of incorrect predictions on the development set by the model without co-attention. The full model makes the correct predictions in these examples.

co-attention helps to identify the undesired utterance with respect to the dialogue history topic. It also manages to make better predictions in the cases where the target system response is completely irrelevant with respect to the the dialogue history.

5.3 Error Analysis

We also perform an error analysis on 100 randomly selected examples from the development set where CXM-D fails to make correct predictions. We identify the following three primary cases where our model tends to make incorrect predictions.

Continuous questions The target utterance is a question and the dialogue history involves continuous question turns where the user and system take turns to ask questions.

Sarcasm We observe that it is challenging for the model to distinguish an undesired response from a sarcastic but appropriate response. In these cases, our model is most likely to classify them as Possible Breakdown (PB).

Overly long responses While the target utterance consists of multiple sentences, it is also challenging to capture how they are interacting with the dialogue history.

6 Related Work

Several prior works on the dialogue breakdown detection task are based on long short-term memory (LSTM). Hendriksen et al. (2019) incorporated LSTM with pretrained word embeddings, such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Wang et al. (2019) followed a similar idea but added convolutional neural network (CNN) to perform textual feature extraction. Shin et al. (2019) utilized bidirectional LSTM and self-attention layers to incorporate the dialogue in representation learning. Prior works also include feature-engineered models. Wang et al. (2019) proposed to utilize a curated set of features including keyword counts and TF-IDF scores to build a regression model based on random forests.

Pretrained language models achieve state-of-the-art performance on many NLP tasks. Sugiyama (2019) developed a model based on BERT (Devlin et al., 2019). Instead of directly using the `[CLS]` token, the model took the concatenation of the entire dialogue including the target system utterance, dialogue acts, and textual features as input to a BERT encoder and utilized the entire last layer for representation learning.

Among the participating models in DBDC4, none of the prior models achieve dominance on both classification-based and distribution-based metrics in either the Japanese or English track. This indicates that it is challenging for a single model to perform well on all metrics. The desired model should possess the capability to alleviate the mismatch between the training objective of classification-based metrics and

distribution-based metrics.

Pretrained language models have been employed in several dialogue tasks and show good performance. BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019) have been utilized to encode system and user utterances in dialogue state tracking (Gao et al., 2019; Ma et al., 2020; Lee et al., 2019). Liu et al. (2020) also incorporated BERT-based contextualized word embeddings for dialogue generation in chat-oriented dialogue systems.

Recently, cross-lingual transfer learning has gained much popularity in the NLP research community (Chen et al., 2019). Cross-lingual pretrained language models are able to map words from different languages to one single shared embedding space. Empirical results show that cross-lingual language models trained on one hundred languages outperform language-specific language models on several standard NLP benchmarks (Conneau and Lample, 2019; Conneau et al., 2020). However, cross-lingual transfer learning has not been explored in dialogue breakdown detection, and our work is the first to incorporate a pretrained cross-lingual language model for dialogue breakdown detection. On the one hand, we use cross-lingual word embeddings to transfer the knowledge learned from multiple languages. On the other hand, as different languages can be mapped to a single shared space in XLM-R (Conneau et al., 2020), we further enrich the training data by adding available data in other languages.

Co-attention network has been used before to tackle the task of single-turn QA (Lu et al., 2016; Xiong et al., 2017; Yu et al., 2019). It shows good capability in capturing the interaction between the context passage and the question in reading comprehension-based QA. In visual QA, it captures the interaction between the compressed image representation and the question. In contrast, we treat the dialogue history and the target system utterance as the two components which interact with each other. The co-attention encoder attends to the two interacting components simultaneously and finally combines both attention contexts.

7 Conclusion

In this paper, we have proposed a novel model based on a cross-lingual language model and a co-attention network for dialogue breakdown detection. Our model achieves new state-of-the-art scores in Dialogue Breakdown Detection Challenge 4. Our proposed model is the first to achieve the best scores on all the evaluation metrics, significantly outperforming all previous models. We have also observed that our model outperforms previous monolingual models on this task. We exploit transfer learning using a cross-lingual language model to utilize training data from other languages and further improve the performance of our model on this task. The co-attention network built on top of the cross-lingual language model better captures the relation between the current utterance and the dialogue history. This helps to reduce the probability of a system generating an undesired response, so that communication and user experience are further improved.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-007). The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore (https://www.nscc.sg).

References

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. 2019. Multi-source cross-lingual model transfer: Learning what to share. In *ACL*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *NeurIPS*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *EMNLP*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *SIGDIAL*.

Mariya Hendriksen, Artuur Leeuwenberg, and Marie-Francine Moens. 2019. LSTM for dialogue breakdown detection: Exploration of different model types and word embeddings. In *IWSDS Workshop*.

Ryuichiro Higashinaka, Kotaro Funakoshi, Yuka Kobayashi, and Michimasa Inaba. 2016. The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics. In *LREC*.

Ryuichiro Higashinaka, Luis Fernando D'Haro, Bayan Abu Shawar, Rafael Enrique Banchs, Kotaro Funakoshi, Michimasa Inaba, Yuiko Tsunomori, Tetsuro Takahashi, and João Sedoc. 2019. Overview of the dialogue breakdown detection challenge 4. In *IWSDS Workshop*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *ACL*.

Qian Liu, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. 2020. You impress me: Dialogue generation via mutual persona perception. In *ACL*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *NeurIPS*.

Yue Ma, Zengfeng Zeng, Dawei Zhu, Xuan Li, Yiying Yang, Xiaoyuan Yao, Kaijie Zhou, and Jianping Shen. 2020. An end-to-end dialogue state tracking system with machine reading comprehension and wide & deep classification. In *AAAI Workshop on Dialog System Technology Challenge*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*.

JongHo Shin, Alireza Dirafzoon, and Aviral Anshu. 2019. Context-enriched attentive memory network with global and local encoding for dialogue breakdown detection. In *IWSDS Workshop*.

Hiroaki Sugiyama. 2019. Dialogue breakdown detection using BERT with traditional dialogue features. In *IWSDS Workshop*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP Workshop on BlackboxNLP*.

Chih-hao Wang, Sosuke Kato, and Tetsuya Sakai. 2019. RSL19BD at DBDC4: Ensemble of decision tree-based and LSTM-based models. In *IWSDS Workshop*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. In *ICLR*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *CVPR*.