A Semi-supervised Learning Approach with Two Teachers to Improve Breakdown Identification in Dialogues

Qian Lin, Hwee Tou Ng

Department of Computer Science, National University of Singapore
qlin@u.nus.edu, nght@comp.nus.edu.sg

Abstract

Identifying breakdowns in ongoing dialogues helps to improve communication effectiveness. Most prior work on this topic relies on human annotated data and data augmentation to learn a classification model. While quality labeled dialogue data requires human annotation and is usually expensive to obtain, unlabeled data is easier to collect from various sources. In this paper, we propose a novel semi-supervised teacher-student learning framework to tackle this task. We introduce two teachers which are trained on labeled data and perturbed labeled data respectively. We leverage unlabeled data to improve classification in student training where we employ two teachers to refine the labeling of unlabeled data through teacher-student learning in a bootstrapping manner. Through our proposed training approach, the student can achieve improvements over single-teacher performance. Experimental results on the Dialogue Breakdown Detection Challenge dataset DBDC5 and Learning to Identify Follow-Up Questions dataset LIF show that our approach outperforms all previous published approaches as well as other supervised and semi-supervised baseline methods.

Introduction

In recent years, interactive virtual conversational agents have been developed rapidly and used widely in daily lives. The information exchange between a user and an agent is done via a conversational dialogue. To achieve effective communication, the agent is expected to generate a proper and rational response based on not only the last turn but also all previous utterances in the dialogue history to continue the dialogue. The user's trust in the agent is damaged when the agent fails to identify the user's intent and generates an inappropriate response, which confuses the user and causes a breakdown in the dialogue. Therefore, identifying breakdowns in dialogues is essential for improving the effectiveness of conversational agents, so that the agent is able to avoid generating responses which cause the breakdowns.

Much prior work on breakdown identification in dialogues has focused on supervised learning on human annotated data. One line of work relies on feature-engineered machine learning methods including decision trees and random forests (Wang, Kato, and Sakai 2019). Another line of work utilizes non-Transformer based neural networks such as LSTM (Hendriksen, Leeuwenberg, and Moens 2019; Wang, Kato, and Sakai 2019; Shin, Dirafzoon, and Anshu 2019). Transformer-based methods involve pre-trained language models which are pre-trained on large corpora (Devlin et al. 2019; Conneau et al. 2020). Sugiyama (2019) and utilize BERT with input consisting of the text and textual features from the dialogue. Lin, Kundu, and Ng (2020) introduce multilingual transfer learning through a cross-lingual pre-trained language model and co-attention modules to reason between the dialogue history and the last utterance.

A recent work (Ng et al. 2020) proposes to perform pre-training on BERT with conversational data and apply self-supervised data augmentation on labeled data. Although good performance has been reported, we observe that the gain of either continued pre-training or data augmentation on labeled data is marginal over the conventional BERT classification scheme. Moreover, pre-training of a pre-trained language model on large corpora is resource-intensive.

We believe that training with dialogue data from other sources introduces diversity and enables the trained model to generalize better. Since annotated dialogue data is expensive to obtain, we propose using unlabeled data through semi-supervised learning and self-training, such that the training data is enriched and more diverse.

In this work[1], we propose a novel semi-supervised teacher-student learning framework to improve the performance of pre-trained language models with unlabeled data. We leverage unlabeled data from other sources to enrich the training set through self-training, which is a general case of domain adaptation where source data and target data are sampled from different data sources. Self-training uses a trained classifier to assign label score vectors on unlabeled data instances. However, such labeling process tends to generate labels under the assumption that a similar distribution is shared by labeled data and unlabeled data. Since the distribution of unlabeled data is difficult to estimate, we introduce two teachers to improve the labeling of unlabeled data. The student model is encouraged to integrate the knowledge from two teachers in a bootstrapping manner.

We leverage a data augmentation method (Yavuz et al. 2020) incorporating [MASK] tokens derived from a Masked

[1]The source code and trained models of this paper are available at https://github.com/nusnlp/S2T2.

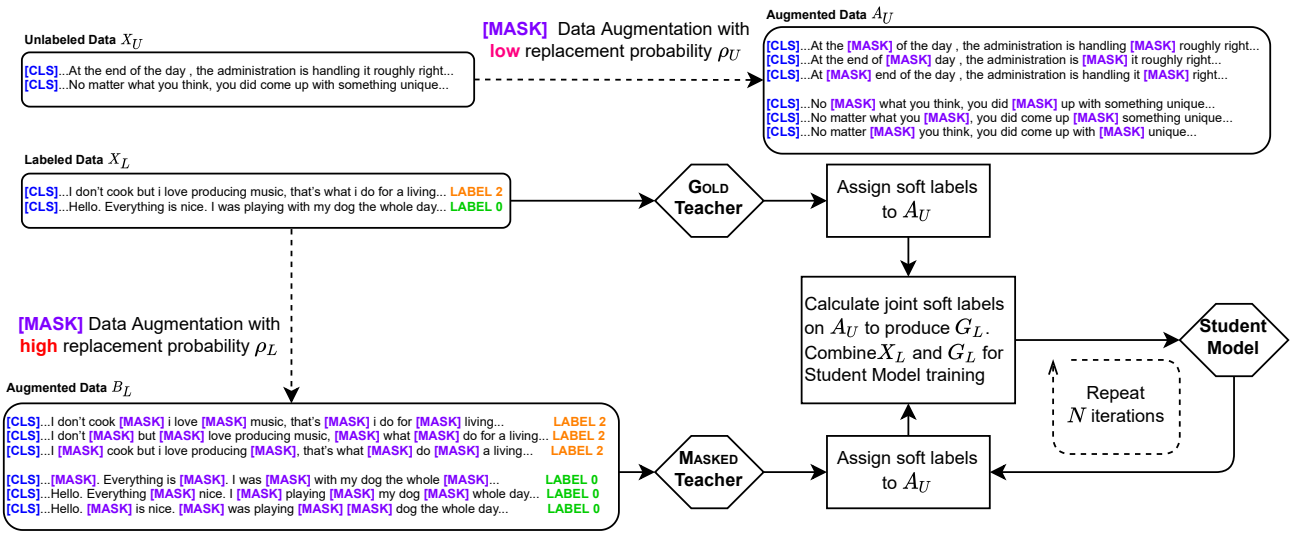

Figure 1: Overview of the proposed training process.

Language Model (MLM) pre-training objective of pre-trained language models (PLM). It is a natural fit to incorporate such augmented data with PLM like RoBERTa (Liu et al. 2019) and XLM-R (Conneau et al. 2020), since the PLMs have adapted to the masking patterns during the pre-training process on large corpora. The GOLD teacher learns knowledge from only labeled data and it tends to generate labels following the distribution of labeled data. The MASKED teacher is trained with only perturbed labeled data which is augmented by randomly replacing tokens from the labeled data with the [MASK] tokens based on a predefined probability.

We construct the training data for the student model as the combination of two segments: labeled data and [MASK]-perturbed unlabeled data. We explicitly impose a difference between masking probabilities applied to labeled data (for training the MASKED Teacher) and unlabeled data (for training the student). Two teachers can provide proper distribution estimation on these two data segments separately. Therefore the student is optimized to distill the knowledge from the two teachers to improve self-training on the combined training set.

We evaluate our proposed approach on two multi-turn dialogue breakdown detection datasets and a large-scale follow-up question identification dataset. Experimental results show that our semi-supervised teacher-student learning framework outperforms all previous published approaches and competitive supervised and semi-supervised baselines. We also conduct further analysis to verify the effectiveness of the training strategies proposed in our framework.

Task Overview

Given a dialogue history $\mathcal{H}$ consisting of a sequence of alternating user and system utterances and the succeeding target system utterance $\mathcal{T}$, the task is to determine whether or not the target utterance causes a certain dialogue breakdown type. Each instance $(\mathcal{H}, \mathcal{T})$ is associated with a soft label vector $\mathbf{y} \in \mathbb{R}^{|C|}$ which corresponds to the probability distribution over the set C covering all possible breakdown types.

Proposed Approach

We give a detailed description of the proposed approach in this section.

Pre-trained Language Model

Assume that the dialogue history $\mathcal{H} = [\mathcal{H}_1, ..., \mathcal{H}_h]$ consists of h tokens after tokenization and the target system utterance $\mathcal{T} = [\mathcal{T}_1, ..., \mathcal{T}_t]$ has t tokens. We first obtain a sequence of tokens by concatenating the two sequences with [CLS] and [SEP] tokens. The input to the pre-trained language model is:

$$x = [\text{CLS}]\mathcal{H}_1...\mathcal{H}_h[\text{SEP}]\mathcal{T}_1...\mathcal{T}_t \quad (1)$$

The combined sequence x includes n tokens. x is first converted to embedding $\mathbf{x}$ by the PLM embedding layer. The output of the pre-trained language model is a sequence of hidden states from the last layer of the model:

$$f(\mathbf{x}; \theta_m) = \text{PLM}(\mathbf{x}) \quad (2)$$

where θ_m denotes the parameters of PLM, d is the hidden size of the pre-trained language model and the output shape is $n \times d$.

Data Augmentation with [MASK] Tokens

We leverage the pre-trained language models with Masked Language Model (MLM) training objective (Devlin et al. 2019; Liu et al. 2019; Conneau et al. 2020) for data augmentation with [MASK] tokens.

We perform data augmentation on the available data by randomly replacing the tokens in the instances with [MASK] tokens (Yavuz et al. 2020).

For each labeled or unlabeled data instance, we generate a certain number of new data instances with [MASK] tokens

based on a predefined replacement probability ρ. For instance, we replace 10% of tokens with [MASK] tokens with $\rho = 0.1$.

For the unlabeled data $X_U = \{(\mathbf{x}_i, \cdot)\}_{i=1}^{|X_U|}$, we eventually obtain the augmented data $\{(\mathbf{x}'_j, \cdot)\}_{j=1}^{k|X_U|}$, where k is the number of augmented instances per original instance. Similarly, for the labeled data $X_L = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{|X_L|}$, we obtain the augmented dataset $\{(\mathbf{x}'_j, \mathbf{y}'_j)\}_{j=1}^{k|X_L|}$ where the label of the augmented instance remains the same as the original instance. We use subscript L to indicate that the dataset is labeled and use subscript U for an unlabeled dataset.

Teacher Models

We introduce two teacher models, namely Gold Teacher (GT) and MASKED Teacher (MT). Both teacher models share the same neural network architecture. Specifically, GOLD Teacher is trained with labeled dataset (gold data) while MASKED Teacher is trained with [MASK] augmented data.

A teacher model is formed by a pre-trained language model and a classification layer. We denote parameters of the teacher model as $\theta^{(\mathrm{T})} = \{\theta_m^{(\mathrm{T})}, \theta_c^{(\mathrm{T})}\}$ where $\theta_m^{(\mathrm{T})}$ denotes parameters of the pre-trained language model and $\theta_c^{(\mathrm{T})}$ the parameters of the classification layer. We use the output at the first position ([CLS]) as the representation of the input $\mathbf{x}$:

$$\mathbf{h} = f(\mathbf{x}; \theta_m^{(\mathrm{T})})[0] \tag{3}$$

The classification layer consists of two linear functions connected by $\tanh$ activation.

$$\mathbf{g} = \mathbf{W}_2(\tanh(\mathbf{W}_1\mathbf{h} + \mathbf{b}_1)) + \mathbf{b}_2 \tag{4}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d\times d}$, $\mathbf{h}, \mathbf{b}_1 \in \mathbb{R}^d$, $\mathbf{W}_2 \in \mathbb{R}^{|C|\times d}$ and $\mathbf{g}, \mathbf{b}_2 \in \mathbb{R}^{|C|}$. The prediction is calculated by:

$$\hat{\mathbf{y}} = f(\mathbf{x}; \theta^{(\mathrm{T})}) = \mathrm{softmax}(\mathbf{g}) \tag{5}$$

The loss function is a weighted sum of three objectives: cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$, supervised contrastive learning loss $\mathcal{L}_{\mathrm{SCL}}$ (Gunel et al. 2021), and mean squared error loss $\mathcal{L}_{\mathrm{MSE}}$.

$$\mathcal{L} = \beta_1\mathcal{L}_{\mathrm{CE}} + \beta_2\mathcal{L}_{\mathrm{SCL}} + \beta_3\mathcal{L}_{\mathrm{MSE}} \tag{6}$$

The supervised contrastive learning loss is defined as:

$$\mathcal{L}_{\mathrm{SCL}} = \sum_{i=1}^{N_b} -\frac{1}{N_{b,y_i} - 1} \sum_{j=1}^{N_b} \mathbb{1}_{i\neq j}\mathbb{1}_{y_i=y_j} \log\frac{\exp\left(\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)/\tau\right)}{\sum_{k=1}^{N_b}\mathbb{1}_{i\neq k}\exp\left(\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_k)/\tau\right)} \tag{7}$$

where $y_i = \mathrm{argmax}(\mathbf{y}_i)$, N_b is the batch size, and N_{b,y_i} is the number of instances with the same label as the i-th instance within the batch. $\mathbb{1}$ denotes the indicator function. τ is a temperature parameter. $\Phi(\cdot)$ corresponds to the encoder function described in Eqn. 3.

We fine-tune all parameters in $\theta^{(\mathrm{T})}$. We use $\theta^{(\mathrm{GT})}$ and $\theta^{(\mathrm{MT})}$ for Gold Teacher and MASKED Teacher respectively.

GOLD Teacher The GOLD Teacher is fine-tuned on labeled dataset $X_L = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{|X_L|}$. It learns knowledge purely from quality annotated data.

Given the unlabeled dataset $X_U = \{(\mathbf{x}_i, \cdot)\}_{i=1}^{|X_U|}$, we augment X_U to $A_U = \{(\mathbf{x}'_i, \cdot)\}_{i=1}^{k|X_U|}$ with k augmented instances per original unlabeled instance and the [MASK] token replacement probability ρ_U.

We use the fine-tuned GOLD Teacher to assign soft labels to the augmented unlabeled dataset A_U.

MASKED Teacher Given the labeled dataset $X_L = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{|X_L|}$ which is the training data for GOLD Teacher, we prepare the training data for MASKED Teacher by generating an augmented dataset on X_L.

The [MASK] augmentation on X_L is determined by k and the [MASK] token replacement probability ρ_L which results in the augmented dataset $B_L = \{(\mathbf{x}'_i, \mathbf{y}'_i)\}_{i=1}^{k|X_L|}$. The MASKED Teacher is fine-tuned on B_L.

The MASKED Teacher learns from training data with [MASK] tokens which adapts to the situation of predicting labels on instances with [MASK] tokens.

We use the fine-tuned MASKED Teacher to assign soft labels to unlabeled instances from A_U which is the same augmented unlabeled dataset described for Gold Teacher.

We set [MASK] token replacement probability ρ_L to be larger than ρ_U ($\rho_L > \rho_U$) such that the MASKED Teacher is robust to produce more confident label scores on A_U.

Student Model

The student model follows the same architecture as the teacher model, consisting of a pre-trained language model and a classification layer.

The parameters of the student model are denoted as $\theta^{(\mathrm{S})} = \{\theta_m^{(\mathrm{S})}, \theta_c^{(\mathrm{S})}\}$ which correspond to the pre-trained language model and the classification layer. The pre-trained language model inherits the weights from the GOLD Teacher fine-tuned on X_L, that is, $\theta_m^{(\mathrm{S})} := \theta_m^{(\mathrm{GT})}$. We do not perform fine-tuning of $\theta_m^{(\mathrm{S})}$ during the training of the student model.

The training objective of the student model is the same as the teacher model, which is defined in Eqn. 6.

Training Process

As mentioned in the last section, the GOLD Teacher is fine-tuned on the labeled dataset X_L and the MASKED Teacher is fine-tuned on B_L where B_L is augmented based on X_L. We also have the unlabeled dataset A_U which is augmented based on the unlabeled dataset X_U. We present the overall training process in Figure 1.

Joint Scoring on Unlabeled Data

For each $\mathbf{x}' \in A_U$, we assign label score vectors by both the GOLD Teacher and the MASKED Teacher respectively.

$$\hat{\mathbf{y}}_{\mathrm{GT}} = f(\mathbf{x}'; \theta^{(\mathrm{GT})}) \tag{8}$$

$$\hat{\mathbf{y}}_{\mathrm{MT}} = f(\mathbf{x}'; \theta^{(\mathrm{MT})}) \tag{9}$$

English	Train / Dev / Unlabeled / Test
#instances	2,110 / 1,950 / 6,150 / 1,940
Japanese	Train / Dev / Unlabeled / Test
#instances	10,418 / 2,818 / 11,506 / 2,672

Table 1: Statistics of DBDC5 English and Japanese datasets.

LIF	Train/Dev/Test-I/Test-II/Test-III
#instances	126,632/5,861/5,992/5,247/2,685
#unlabeled	101,448

Table 2: Statistics of LIF dataset.

We define a joint scoring function to compute the label score vector with weights determined by hyperparameter γ.

$$\hat{\mathbf{y}} = \gamma \hat{\mathbf{y}}_{\text{GT}} + (1-\gamma)\hat{\mathbf{y}}_{\text{MT}} \tag{10}$$

We then obtain the labeled set $G_L = \{(\mathbf{x}'_i, \mathbf{y}'_i)\}_{i=1}^{|A_U|}$ where $\mathbf{x}' \in A_U$ and $\mathbf{y}'$ is calculated by Eqn. 10.

Bootstrapping Strategy

We consider a bootstrapping strategy to refine the labeling of unlabeled data to improve classification. Continuing with G_L obtained by the process mentioned in the last subsection, we describe the bootstrapping strategy for student model fine-tuning as follows.

We use the combined dataset $X_L^{(\text{S})} = X_L \cup G_L$ as the initial training set for the student model. After a complete training iteration consisting of k_e epochs, the fine-tuned student model predicts label score vectors for each unlabeled data $\mathbf{x}' \in A_U$.

$$\hat{\mathbf{y}}_{\text{S}} = f(\mathbf{x}'; \theta^{(\text{S})}) \tag{11}$$

The refined label score vector after each training iteration is calculated by:

$$\begin{aligned} &\lambda = i/N \quad (i = 1, 2, ..., N-1) \\ &\hat{\mathbf{y}} = \alpha[(1+\lambda)\hat{\mathbf{y}}_{\text{S}} + (1-\lambda)\hat{\mathbf{y}}_{\text{MT}}] \end{aligned} \tag{12}$$

where i is the iteration index and N the total number of iterations. Therefore, λ ranges between 0 and 1 ($0 < \lambda < 1$). We set $\alpha = 0.5$ in our experiments, such that we ramp up the weight of $\hat{\mathbf{y}}_{\text{S}}$ from 0.5 to 1.0 while progressively decreasing the contribution from the MASKED Teacher to produce better predictions for the unlabeled data.

As a result, the label score vectors in G_L and the combined training set $X_L^{(\text{S})}$ are updated after each training iteration.

For training of the student model in the succeeding iteration, we retain the parameters from the best epoch in the last iteration based on development set performance. We use the trained student model to make predictions on test sets.

Experiments

Datasets

We evaluate our proposed approach on two multi-turn dialogue datasets DBDC5 English Track (Higashinaka et al. 2020) and DBDC5 Japanese Track (Higashinaka et al. 2020)(Higashinaka et al. 2020), and one much larger Learning to Identify Follow-Up Questions dataset LIF (Kundu, Lin, and Ng 2020).

DBDC5 English Track This is a multi-turn dialogue dataset which requires identification of the predefined dialogue breakdown type of the last system utterance given the dialogue history. Based on the annotation quality (Higashinaka et al. 2020), we use the re-annotated DBDC4 data as the labeled dataset. For the unlabeled data, we use the English data released in Higashinaka et al. (2017).

DBDC5 Japanese Track This is a Japanese dataset with the same format as DBDC5 English Track. For the Japanese track, we use datasets released in previous DBDC tasks as training set, including DBDC1, DBDC2, DBDC3, and DBDC4 development sets, as well as DBDC5 development set. We use DBDC4 evaluation set for validation. These data were annotated by 15–30 annotators per instance. We use Chat dialogue corpus as the source of unlabeled data, which were annotated by only 2–3 annotators. (Higashinaka et al. 2019)

LIF LIF (Kundu, Lin, and Ng 2020) is a conversational question answering dataset for the task of follow-up question identification, which requires the model to identify whether or not the last question follows up on the context passage and previous conversation history. Since LIF is derived from QuAC (Choi et al. 2018), we select the training set of CoQA (Reddy, Chen, and Manning 2019) which is a similar conversational QA dataset, as the source of unlabeled data.[2]

We present the statistics of both DBDC5 datasets in Table 1 and the statistics of the LIF dataset in Table 2. The numbers reported do not include augmented data.

Evaluation Metrics

DBDC5 English Track and DBDC5 Japanese Track require classification-based metrics including accuracy and F1 scores, and distribution-based metrics including Jensen-Shannon divergence (JSD) and Mean Squared Error (MSE).[3]

LIF dataset requires classification-based metrics including precision, recall, and F1 of class *Valid*, and macro F1.[4]

Experimental Setup

We experiment with RoBERTa (Liu et al. 2019) as the pre-trained language model in DBDC5 English Track and LIF. We use multilingual pre-trained language model XLM-R (Conneau et al. 2020) for DBDC5 Japanese Track. We use the large version with hidden size $d = 1024$. The maximum input length is set to 256.

For experiments on LIF, we concatenate the context passage and the conversation history into $\mathcal{H}$, and $\mathcal{T}$ corresponds to the candidate question.

The weights β_1, β_2, β_3 are set to 1e-2, 1e-3, and 1.0 in the loss function for both DBDC5 English Track and Japanese

[2] As we use CoQA samples without modification, the samples do not include the cases where the candidate question is from other conversations (Kundu, Lin, and Ng 2020), we suggest these samples still contribute to the generalization.

[3] Refer to Higashinaka et al. (2020) for details.

[4] Refer to Kundu, Lin, and Ng (2020) for details.

Model	English				Japanese			
	Accuracy	F1(B)	JSD↓	MSE↓	Accuracy	F1(B)	JSD↓	MSE↓
BERT+SSMBA	0.739	0.782	0.070	0.036	–	–	–	–
XLMR+CM	–	–	–	–	0.745	0.694	0.077	0.040
$\text{PLM}_\flat$ Baseline$^\#$	0.721	0.765	0.080	0.043	0.706	0.659	0.084	0.044
PLM Baseline	0.750	0.797	0.066	0.033	0.732	0.650	0.074	0.039
PLM+CoAtt	0.752	0.794	0.067	0.033	0.740	0.708	0.069	0.035
RoBERTa+SSMBA	0.752	0.797	0.067	0.035	–	–	–	–
UDA	0.754	0.799	0.071	0.037	0.733	0.692	0.075	0.039
MixText	0.757	0.805	0.059	0.030	0.743	0.715	0.073	0.036
Ours (X_U, no A_U)	0.759	0.803	0.065	0.033	0.747	0.721	0.068	0.036
Ours	**0.779**	**0.824**	**0.058**	**0.028**	**0.767**	**0.754**	**0.062**	**0.031**

Table 3: Experimental results on the DBDC5 English and Japanese track. ↓ the lower the better. $\#$ subscript $\flat$ denotes the base version of PLM.

Models	**Test-I** V-P/-R/-F1/Macro F1	**Test-II** V-P/-R/-F1/Macro F1	**Test-III** V-P/-R/-F1/Macro F1
Three-way AP	74.4/75.7/75.0/81.4	89.0/75.7/81.8/86.2	81.9/75.7/78.7/65.0
PLM Baseline	75.6/85.4/80.2/84.9	88.2/85.4/86.8/89.6	84.2/85.4/84.8/72.0
PLM+CoAtt	76.6/80.6/78.5/83.9	87.7/80.6/84.0/87.6	85.8/80.6/83.1/71.8
UDA	**79.2**/83.9/81.5/86.1	**91.4**/83.9/87.5/90.3	85.6/83.9/84.7/73.2
MixText	74.8/86.6/80.3/84.8	87.8/86.6/87.2/89.9	83.4/86.6/85.0/71.5
Ours (X_U, no A_U)	78.0/83.9/80.9/85.6	89.5/83.9/86.6/89.6	**85.9**/83.9/84.9/73.5
Ours (4.74%*)	73.9/83.3/78.3/83.5	87.5/83.3/85.4/88.6	82.6/83.3/83.0/69.0
Ours (25%*)	74.6/86.9/80.3/84.8	88.7/86.9/87.8/90.4	82.4/86.9/84.6/70.1
Ours (50%*)	77.6/**87.3**/82.1/86.4	89.3/**87.3**/88.3/90.8	85.5/**87.3/86.4/74.8**
Ours (100%*)	78.1/86.6/**82.2/86.5**	90.6/86.6/**88.6/91.0**	85.0/86.6/85.8/73.8

Table 4: Experimental results on the LIF dataset. V-P, V-R, and V-F1 correspond to precision, recall, and F1 score on class *Valid*. $\star$ denotes the percentage of the LIF training dataset used.

Track. Since LIF does not require distribution-based metrics, the weights $\beta_1, \beta_2, \beta_3$ are set to 1.0, 0.1, and 0 in experiments on LIF. We set temperature τ to 1.0 in $\mathcal{L}_{\text{SCL}}$ and γ to 0.5 in Eqn. 10. We optimize the loss using AdamW (Loshchilov and Hutter 2019) with 0.01 weight decay.

For data augmentation, we generate 6 instances for each labeled or unlabeled instance. We set [MASK] token replacement probability $\rho_U = 0.15$ aligning to (Devlin et al. 2019; Liu et al. 2019) and $\rho_L = 0.25$.

To train two teacher models, we use a batch size of 16, 8, and 12 for experiments on DBDC5 English Track, DBDC5 Japanese Track, and LIF respectively. The learning rate during training is set to 1e-5, 1e-5, and 2e-6 respectively. To train the student model, we use a batch size of 128 and learning rate 2e-6 for experiments on all three datasets. We set the maximum number of iterations N to 5 and the number of epochs k_e to 5 per iteration. Models are trained on a single Tesla V100 GPU.

Compared Models

BERT+SSMBA (Ng et al. 2020) The model consists of pre-trained language model BERT-base and a classification layer. The BERT parameters are further pretrained on large-scale Reddit dataset. The labeled training data is augmented based on SSMBA (Ng, Cho, and Ghassemi 2020) and original labels are assigned to augmented instances. This is the best-performing model published to date on the DBDC5 English track. We implement a baseline **RoBERTa+SSMBA** using RoBERTa classification model with SSMBA augmentation. For fairer comparison with our proposed approach on DBDC5 English dataset, we adopt RoBERTa-large model and generate data with SSMBA which follows BERT+SSMBA.

XLMR+CM The model uses cross-lingual language model XLM-R with context matching (CM) modules. This is the best-performing model published to date on the DBDC5 Japanese track (Higashinaka et al. 2020).

Three-way AP (Kundu, Lin, and Ng 2020) The model applies an attentive pooling network to capture interactions among the context passage, conversation history, and the candidate follow-up question. This is the best performing-model published to date on the LIF dataset.

PLM Baseline We build a simple but effective baseline model consisting of a pre-trained language model and a classification layer. We select RoBERTa for experiments on English tasks (DBDC5 English Track and LIF) and XLM-R for experiments on DBDC5 Japanese track. We use the output from [CLS] as the representation for classification. We adopt the large version of the pre-trained language model (RoBERTa-large or XLM-R-large) unless stated otherwise.

PLM+CoAtt We build another baseline by applying a co-attention network (Lin, Kundu, and Ng 2020) on the output from a pre-trained language model. The selection of pre-trained language models follows **PLM Baseline**. Since the co-attention network applies to two sequences of representations

corresponding to conversation history and the last utterance, we prepend the context passage to the conversation history and the candidate question is treated as the last utterance for experiments on LIF.

We also experiment with recently proposed semi-supervised methods **UDA** (Xie et al. 2020) and **MixText** (Chen, Yang, and Yang 2020). We adopt RoBERTa-large for the English datasets and XLM-R-large for the Japanese dataset. Labeled and unlabeled data (before augmentation) used are the same as our proposed method. For English datasets DBDC5 English and LIF, we use back-translation with German and Russian as intermediate languages for augmentation on unlabeled data following Chen, Yang, and Yang (2020). For DBDC5 Japanese dataset, we apply [MASK] augmentation used in our proposed method due to the non-availability of Japanese round-trip back-translation model. UDA and MixText are trained on both labeled and unlabeled data, while the other compared models are trained on labeled data in a supervised manner.

Results

Main Results

We present the experimental results of DBDC5 (both English Track and Japanese Track) and LIF in Table 3 and Table 4, respectively. Results of BERT+SSMBA and XLMR+CM are retrieved from Higashinaka et al. (2020) and results of Three-way AP are retrieved from Kundu, Lin, and Ng (2020). For results of both DBDC5 datasets, we report Accuracy, F1(B), JS Divergence (JSD), and Mean Squared Error (MSE). For metrics MSE and JSD, the reported percentage of improvement is calculated as $100 - (ours/other\_model) \times 100$.

In the DBDC5 English Track, our proposed approach outperforms the prior best-performing model (BERT+SSMBA) by 4.0%, 4.2%, 17.1%, and 22.2% on metrics Accuracy, F1(B), JSD, and MSE. It also performs better than all supervised and semi-supervised baselines by at least 2.2%, 1.9%, 1.7%, and 6.7% on the reported four metrics. The results of RoBERTa+SSMBA show that based on the large pre-trained language model setting, adding SSMBA augmented data does not contribute improvement on this task.

In the DBDC5 Japanses Track, our proposed approach outperforms the prior best-performing model (XLMR+CM) by 2.2%, 6.0%, 19.5%, and 22.5% on metrics Accuracy, F1(B), JSD, and MSE. The improvements are at least 2.4%, 3.9%, 10.1%, and 11.4% when compared to all other baseline models. We notice that models with large version of PLM perform generally better on these datasets.

In the much larger LIF dataset, we sample different sizes of labeled training data from the full training dataset to verify the robustness of our approach. The smallest sampled training set consists of only 6,000 (4.74% of 126,632) labeled training instances, similar to the sizes of Test-I and Test-II. In this case, we sample 18,000 instances from CoQA as unlabeled data and increase the sample size accordingly for the larger training sets. With only 6,000 labeled training instances, our method achieves competitive performance which outperforms the previous best-performing model (Three-way AP) on all three LIF test sets except V-P of Test-I and Test-II.

Model	D-EN	D-JP	LIF
Ours (full)	**77.9**	**76.7**	**86.5**
– MT	76.9	75.9	86.1
– self-training	75.0	73.2	84.9
– GT	74.0	73.4	85.9
– self-training	73.8	73.1	84.8

Table 5: Performance on the test set after removing (–) different components. We report Accuracy on DBDC5 English (D-EN) and DBDC5 Japanese (D-JP) and Macro F1 on LIF Test-I. GT: GOLD Teacher. MT: MASKED Teacher.

We also experiment with 25%, 50%, and 100% of full training data and observe further performance improvement. With 100% labeled training data, our approach outperforms Three-way AP on all metrics by a wide margin and also performs better than other supervised and semi-supervised baselines on all metrics except for V-P.

We conduct experiments comparing the use of X_U and A_U when training the student model. We replace A_U as X_U in our original approach and denote this variation as Ours (X_U, no A_U). The results on the test sets indicate that utilizing A_U (augmentations on unlabeled data) is more effective.

We perform statistical significance tests with regards to Accuracy (DBDC5 English and Japanese datasets) and Macro F1 (LIF) on test sets. Our proposed method is significantly better ($p < 0.05$) than all baseline methods.

Analysis

Based on our implementation of teacher and student models where teachers and student use the same architecture, our proposed approach is in line with the idea of self-distillation (Mobahi, Farajtabar, and Bartlett 2020; Furlanello et al. 2018). It has been observed that self-distillation helps to improve test performance (Liu, Shen, and Lapata 2021; Furlanello et al. 2018; Zhang et al. 2019). Allen-Zhu and Li (2020) show that self-distillation performs implicit ensemble with knowledge distillation. In traditional self-distillation, the student is distilled from a single trained teacher. In our approach, we distill the knowledge from two different trained teachers with additional unlabeled data. We evaluate the effectiveness of two teachers by removing components in the proposed approach and show the results in Table 5. Performance drops when we remove either teacher but keep the self-training on combined training set. We observe that performance drops further if we continue to remove the self-training process. This shows that both teachers contribute to the performance improvement on these tasks.

We conduct analysis of different settings on the development set, in order to select hyperparameters as well as to better understand the effectiveness of our proposed approach. For development set performance, we report accuracy score in the DBDC5 English Track and DBDC5 Japanese Track, and Macro F1 score in LIF6000 in which the model is provided with 6,000 labeled training instances sampled from LIF.

We select the number of augmented samples k from $\{4,6,8\}$ and observe that the sample size 6 performs con-

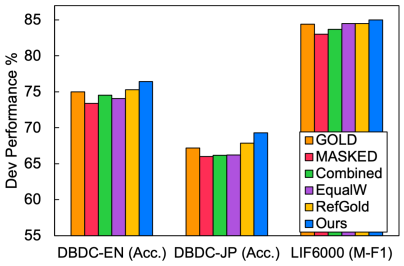

Figure 2: Performance comparison on different training strategies on the development set.

sistently better than the other two on all three datasets.

We investigate the impact of [MASK] token replacement probability of labeled data (ρ_L) on the model performance. Given a constant $\rho_U = 0.15$, we vary the value of ρ_L from 0.15 to 0.30 with step size of 0.05. The best development set performance is achieved at $\rho_L = 0.25$.

We also explore different training strategies and compare them with our proposed approach. **GOLD** denotes that we only use GOLD Teacher which is trained on labeled data X_L only to make predictions. **MASKED** denotes that we only use MASKED Teacher which is trained on [MASK] augmentation of labeled data B_L to make predictions. **Combined** means the model is trained from scratch on the combination of labeled data and [MASK] augmentation of labeled data ($X_L \cup B_L$) without bootstrapping. **EqualW** indicates the training method in which we use trained GOLD Teacher and trained MASKED Teacher to make predictions on unlabeled data and score equally for the final label scores. That is, $\hat{\mathbf{y}} = 0.5\hat{\mathbf{y}}_{\text{GT}} + 0.5\hat{\mathbf{y}}_{\text{MT}}$ in Eqn. 10 for generating labels for unlabeled data and obtaining G_L. We then train the student model on $X_L \cup G_L$ without bootstrapping. **RefGold** denotes a variant of our bootstrapping approach where the score refinement in Eqn. 12 is altered to $\hat{\mathbf{y}} = \alpha[(1+\lambda)\hat{\mathbf{y}}_{\text{S}} + (1-\lambda)\hat{\mathbf{y}}_{\text{GT}}]$ in which we refer to GOLD Teacher. We present performance comparison in Figure 2.

Our proposed approach outperforms all other mentioned training strategies on the development set. **GOLD** achieves the best performance among non-bootstrapping settings, indicating that preserving knowledge from **GOLD** Teacher is important, which validates the initialization of our proposed student model. **RefGold** produces slightly lower scores than our proposed approach, probably because the number of masked training instances is more than instances without [MASK] tokens during bootstrapping training, so using predictions (Eqn. 12) from MASKED Teacher is better. But it still performs better than other non-bootstrapping settings. This finding suggests that the proposed bootstrapping is essential for further performance improvement. Our proposed method (Ours) is also significantly better ($p < 0.05$) than **GOLD** and **RefGold**.

Related Work

Data Augmentation Recent unsupervised data augmentation methods have shown the effectiveness on classification tasks with short text instances. Wei and Zou (2019) introduce random word-level operations including replacement, insertion, deletion, and swapping. Xie et al. (2020) add noise to the unlabeled data and generate new training data by back-translation. These augmentation methods tend to generate unnatural text samples as the text sequence becomes longer such as multi-turn dialogues and conversations. A recent data augmentation method based on self-supervised learning is proposed to tackle the out-of-domain issue (Ng et al. 2020). Another line of recent work proposes to augment data by randomly replacing word tokens with [MASK] tokens while working with pre-trained language models pre-trained with Masked Language Model objective (Yavuz et al. 2020). In our work, we leverage this idea and further investigate how different probabilities of [MASK] token replacement affect the model performance.

Domain Adaptation Domain adaptation is a general method which transfers the knowledge from a source domain to a target domain. Domain adaptation is usually applied to text classification tasks when labeled source domain data is more abundant than target domain data. It enables a classifier trained on a source domain to be generalized to another target domain (Jiang and Zhai 2007; Chen, Weinberger, and Blitzer 2011; Chen et al. 2012). Recent works incorporate output features from pre-trained language models to improve domain adaptation (Nishida et al. 2020; Ye et al. 2020). In our work, we sample unlabeled data from sources other than the available labeled training set to enrich the training data. We leverage the idea of domain adaptation with source data and target data sampled from different data sources.

Semi-supervised Learning In our work, we utilize unlabeled data from other sources for model training via semi-supervised learning. Semi-supervised learning involves both labeled and unlabeled data during training. The general idea is to train a model with labeled data in a supervised learning manner and then enrich the labeled set with the most confident predictions on unlabeled data (Kehler et al. 2004; McClosky, Charniak, and Johnson 2006; Oliver et al. 2018; Li et al. 2019). Regularization techniques are applied to obtain better decision boundaries of unlabeled data with unknown distribution, including adversarial training (Miyato, Dai, and Goodfellow 2017), adding dropout, adding noise, and bootstrapping (Laine and Aila 2017). We consider the bootstrapping strategy to refine the labeling of unlabeled data. Prior work shows that bootstrapping improves the labeling of unlabeled data (Reed et al. 2015; Laine and Aila 2017; He et al. 2018).

Conclusion

In this work, we propose a novel semi-supervised teacher-student learning framework with two teachers. We leverage both labeled and unlabeled data during training in a bootstrapping manner. We show that bootstrapping with the proposed re-labeling method is essential to improve performance. Evaluation results on two multi-turn dialogue breakdown detection datasets and a large-scale follow-up question identification dataset show that our proposed method achieves substantial improvements over prior published methods and competitive baselines.

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

Allen-Zhu, Z.; and Li, Y. 2020. Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning. *CoRR*, arXiv:2012.09816.

Chen, J.; Yang, Z.; and Yang, D. 2020. Mixtext: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In *ACL*, 2147–2157.

Chen, M.; Weinberger, K. Q.; and Blitzer, J. 2011. Co-Training for Domain Adaptation. In *NIPS*, 2456–2464.

Chen, M.; Xu, Z.; Weinberger, K. Q.; and Sha, F. 2012. Marginalized Denoising Autoencoders for Domain Adaptation. In *ICML*, 1627–1634.

Choi, E.; He, H.; Iyyer, M.; Yatskar, M.; Yih, W.-t.; Choi, Y.; Liang, P.; and Zettlemoyer, L. 2018. QuAC: Question Answering in Context. In *EMNLP*, 2174–2184.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *ACL*, 8440–8451.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 4171–4186.

Furlanello, T.; Lipton, Z.; Tschannen, M.; Itti, L.; and Anandkumar, A. 2018. Born Again Neural Networks. In *ICML*, 1607–1616.

Gunel, B.; Du, J.; Conneau, A.; and Stoyanov, V. 2021. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. In *ICLR*.

He, R.; Lee, W. S.; Ng, H. T.; and Dahlmeier, D. 2018. Adaptive Semi-supervised Learning for Cross-domain Sentiment Classification. In *EMNLP*, 3467–3476.

Hendriksen, M.; Leeuwenberg, A.; and Moens, M.-F. 2019. LSTM for Dialogue Breakdown Detection: Exploration of Different Model Types and Word Embeddings. In *IWSDS Workshop*.

Higashinaka, R.; D'Haro, L. F.; Shawar, B. A.; Banchs, R. E.; Funakoshi, K.; Inaba, M.; Tsunomori, Y.; Takahashi, T.; and Sedoc, J. 2019. Overview of the Dialogue Breakdown Detection Challenge 4. In *IWSDS Workshop*.

Higashinaka, R.; Funakoshi, K.; Inaba, M.; Tsunomori, Y.; Takahashi, T.; and Kaji, N. 2017. Overview of Dialogue Breakdown Detection Challenge 3. In *DSTC6 Workshop*.

Higashinaka, R.; Tsunomori, Y.; Takahashi, T.; Tsukahara, H.; Araki, M.; Sedoc, J.; Banchs, R. E.; and D'Haro, L. F. 2020. Overview of Dialogue Breakdown Detection Challenge 5. In *IWSDS Workshop*.

Jiang, J.; and Zhai, C. 2007. Instance Weighting for Domain Adaptation in NLP. In *ACL*, 264–271.

Kehler, A.; Appelt, D.; Taylor, L.; and Simma, A. 2004. Competitive Self-Trained Pronoun Interpretation. In *NAACL*, 33–36.

Kundu, S.; Lin, Q.; and Ng, H. T. 2020. Learning to Identify Follow-Up Questions in Conversational Question Answering. In *ACL*, 959–968.

Laine, S.; and Aila, T. 2017. Temporal Ensembling for Semi-Supervised Learning. In *ICLR*.

Li, X.; Sun, Q.; Liu, Y.; Zhou, Q.; Zheng, S.; Chua, T.-S.; and Schiele, B. 2019. Learning to Self-Train for Semi-Supervised Few-Shot Classification. In *NeurIPS*, 10276–10286.

Lin, Q.; Kundu, S.; and Ng, H. T. 2020. A Co-Attentive Cross-Lingual Neural Model for Dialogue Breakdown Detection. In *COLING*, 4201–4210.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, arXiv:1907.11692.

Liu, Y.; Shen, S.; and Lapata, M. 2021. Noisy Self-Knowledge Distillation for Text Summarization. In *NAACL*, 692–703.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

McClosky, D.; Charniak, E.; and Johnson, M. 2006. Effective Self-Training for Parsing. In *NAACL*, 152–159.

Miyato, T.; Dai, A. M.; and Goodfellow, I. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. In *ICLR*.

Mobahi, H.; Farajtabar, M.; and Bartlett, P. 2020. Self-Distillation Amplifies Regularization in Hilbert Space. In *NeurIPS*, 3351–3361.

Ng, N.; Cho, K.; and Ghassemi, M. 2020. SSMBA: Self-Supervised Manifold Based Data Augmentation for Improving Out-of-Domain Robustness. In *EMNLP*, 1268–1283.

Ng, N.; Ghassemi, M.; Thangarajan, N.; Pan, J.; and Guo, Q. 2020. Improving Dialogue Breakdown Detection with Semi-Supervised Learning. In *NeurIPS Workshop on Human in the Loop Dialogue Systems*.

Nishida, K.; Nishida, K.; Saito, I.; Asano, H.; and Tomita, J. 2020. Unsupervised Domain Adaptation of Language Models for Reading Comprehension. In *LREC*, 5392–5399.

Oliver, A.; Odena, A.; Raffel, C. A.; Cubuk, E. D.; and Goodfellow, I. 2018. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. In *NeurIPS*, 3239–3250.

Reddy, S.; Chen, D.; and Manning, C. D. 2019. CoQA: A Conversational Question Answering Challenge. *TACL*, 249–266.

Reed, S. E.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; and Rabinovich, A. 2015. Training Deep Neural Networks on Noisy Labels with Bootstrapping. In *ICLR Workshop*.

Shin, J.; Dirafzoon, A.; and Anshu, A. 2019. Context-enriched Attentive Memory Network with Global and Local Encoding for Dialogue Breakdown Detection. In *IWSDS Workshop*.

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

Wang, C.-h.; Kato, S.; and Sakai, T. 2019. RSL19BD at DBDC4: Ensemble of Decision Tree-based and LSTM-based Models. In *IWSDS Workshop*.

Wei, J.; and Zou, K. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *EMNLP*, 6382–6388.

Xie, Q.; Dai, Z.; Hovy, E.; Luong, T.; and Le, Q. 2020. Unsupervised Data Augmentation for Consistency Training. In *NeurIPS*, 6256–6268.

Yavuz, S.; Hashimoto, K.; Liu, W.; Keskar, N. S.; Socher, R.; and Xiong, C. 2020. Simple Data Augmentation with the Mask Token Improves Domain Adaptation for Dialog Act Tagging. In *EMNLP*, 5083–5089.

Ye, H.; Tan, Q.; He, R.; Li, J.; Ng, H. T.; and Bing, L. 2020. Feature Adaptation of Pre-Trained Language Models across Languages and Domains with Robust Self-Training. In *EMNLP*, 7386–7399.

Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; and Ma, K. 2019. Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. In *ICCV*, 3713–3722.