Self-supervised Pre-training and Semi-supervised Learning for Extractive Dialog Summarization

Yingying Zhuang
yyzhuang@amazon.com
Amazon
San Francisco, CA, USA

Jiecheng Song
jiecsong@amazon.com
Amazon
San Francisco, CA, USA

Narayanan Sadagopan
sdgpn@amazon.com
Amazon
San Francisco, CA, USA

Anurag Beniwal
beanurag@amazon.com
Amazon
San Francisco, CA, USA

ABSTRACT

Language model pre-training has led to state-of-the-art performance in text summarization. While a variety of pre-trained transformer models are available nowadays, they are mostly trained on documents. In this study we introduce self-supervised pre-training to enhance the BERT model's semantic and structural understanding of dialog texts from social media. We also propose a semi-supervised teacher-student learning framework to address the common issue of limited available labels in summarization datasets. We empirically evaluate our approach on extractive summarization task with the TWEETSUMM corpus, a recently introduced dialog summarization dataset from Twitter customer care conversations and demonstrate that our self-supervised pre-training and semi-supervised teacher-student learning are both beneficial in comparison to other pre-trained models. Additionally, we compare pre-training and teacher-student learning in various low data-resource settings, and find that pre-training outperforms teacher-student learning and the differences between the two are more significant when the available labels are scarce.

CCS CONCEPTS

• **Information systems** → *Content analysis and feature selection.*

KEYWORDS

summarization, twitter, dialog, self-supervised pre-training, semi-supervised learning

ACM Reference Format:
Yingying Zhuang, Jiecheng Song, Narayanan Sadagopan, and Anurag Beniwal. 2023. Self-supervised Pre-training and Semi-supervised Learning for Extractive Dialog Summarization. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3543873.3587680

WWW '23 Companion, April 30–May 04, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9419-2/23/04…$15.00
https://doi.org/10.1145/3543873.3587680

1 INTRODUCTION

Automatic dialog summarization systems are useful in many applications such as summarizing meeting transcripts, media interviews and customer service conversations, where the goal is to create a concise version of large amounts of textual data from a multi-party dialog to help people quickly capture the most important information. The recent outbreak of the COVID-19 pandemic has led to exponential growth of online communications on social networks such as Facebook or Twitter, and there is an urgent need for summarization techniques to distill key information from long dialogs.

Language model pre-training has shown to be beneficial to text summarization as it improves semantic understanding of natural language going beyond the meaning of individual words and sentences. Large-scale pre-trained transformers such as BERT [13] have led to state-of-the-art performance in text summarization [17]. The pre-training of these models is performed on large documents corpora such as Wikipedia or books which are very different from dialog texts. Firstly, documents are single-participant while a dialog contains multiple participants and dynamic interactions between them [7]. Secondly, the documents used for pre-training are non topic or domain-specific, while dialogs are often goal-oriented and domain-specific. These language mismatches can lead to inefficiencies in semantic and structural understanding of dialog texts. In this paper, we introduce self-supervised pre-training to enhance BERT's ability to contextualize text representations specifically for goal-oriented and domain-specific dialogs from posts/replies on social media which are crucial for the downstream task of summarization.

Furthermore, acquiring labeled data for dialog summarization is extremely expensive and time consuming. It requires large amounts of manual annotation to generate dialog-summary pairs so that each dialog has a summary available to supervise the training of a model that can produce such summaries to capture dialog highlights. The lack of large-scale human annotated dialog-summary pairs has become one of the main bottlenecks to build dialog summarization models. To mitigate these challenges we introduce a semi-supervised teacher-student training framework in low-resource data settings where there are small amount of labels while large-scale unlabeled data are available.

The contributions of our work include:

- We present various self-supervised pre-training methods to learn the contextualized dialog text representations, upon which an extractive summarization task can be fine-tuned.
- To the best of our knowledge, we are the first to consider using a teacher-student framework with limited labels for social media dialog summarization.
- Benefiting from both self-supervised pre-training and the teacher-student learning, our model achieves state-of-the-art results in the dialog extractive summarization task on a Twitter dialog dataset.
- We also compare pre-training vs teacher-student learning in various low data-resource settings and find that pre-training surpasses teacher-student learning, especially when labels are scarce.

The paper is structured as follows: we first discuss related work in Section 2 and then describe the self-supervised pre-training strategies to enhance BERT in Section 3. We then introduce the teacher-student framework in Section 4. In Section 5 we describe the datasets, experiments, and implementation details. Section 6 follows with experimental results and discussions. Finally, we conclude the paper with a summary and future work in Section 7.

2 RELATED WORK

Automatic summarization can be mainly divided into two paradigms: *extractive* and *abstractive*. Extractive summarization directly selects and assembles phrases and sentences from the original texts as the summary, which is more accurate and faithful, while abstractive methods generate a summary with novel words, which improves the conciseness and fluency of the summary. Extractive methods have been proven to be effective by many previous works [4, 21, 29, 30]. Extractive summarization can be modeled as a sentence classification problem of predicting which sentences to be included in the summary.

Recently, the introduction of large-scale pre-trained transformers such as BERT and GPT-2 [13, 23] has led to state-of-the-art results in text summarization [17]. Since the initial release of BERT, several pre-trained transformer encoder variations have been published, such as Roberta [18], Electra [5], Albert [15], Reformer [14] and Long-former [2]. Most pre-trained transformer encoders are pre-trained on text corpora such as Wikipedia, News or Books, which do not represent the multiparty conversing structure inherent to a dialog. Conversational-BERT [**?**] is a BERT model trained on several chit-chat type dialog corpora, e.g., Twitter, Reddit, and Facebook Messages that are neither task-oriented nor domain-specific. MPC-BERT [9] is another BERT model trained on multi-party conversations. However, the conversations are from Ubuntu Internet Relay Chat logs, a collection of logs from Ubuntu-related chat rooms where a number of users chat and discuss various topics. These chat logs are again neither task-oriented nor domain- specific.

Self-supervised pre-training is an emerging topic in recent literatures [25, 26, 28]. Zhang et al. [28] adopted masked sentence prediction in pre-training stage for extractive summarization on large-scale news corpus. Wang et al. [25] introduced self-supervised learning to capture document-level context in order to train an extractive summarization with CNN/DM datasets. However, despite much effort being devoted to pre-training transformer based encoders, there is still a mismatch between the pre-training texts and the downstream domain-specific dialogs. Gururangan et al. [10] proposed two pre-training techniques: domain-adaptive pre-training and task-adaptive pre-training. They have studied across different domains and shown that a second phase of pre-training on in-domain datasets leads to performance gains. Additionally, continued pre-training on the task dataset itself or data relevant to the task can be helpful too. We investigate both domain-adaptive pre-training and task-adaptive pre-training on summarization tasks in further details in this paper.

Even with a pre-trained model, thousands of labels are still typically needed for task-specific fine-tuning to reach satisfactory performance. Semi-supervised learning with teacher-student training is one of the promising paradigms to make use of unlabeled data to address the shortcomings of lack of large scale labels. In this framework, a *teacher* model is first trained on the available labeled data and then applied on the unlabeled data to generate pseudo labels. Then a *student* model is trained on the augmented dataset with the original labeled data as well as pseudo labeled data, after which the *teacher* model is re-initialized with the learned weights of the *student*. This training progresses iteratively until convergence. Such framework has been shown to obtain state-of-the-art performance for tasks like neural machine translation [12], name entity recognition [27], and distillation [20]. Yet it has not been explored in limited label settings for dialog summarization.

3 SELF-SUPERVISED PRE-TRAINING FOR DIALOG UNDERSTANDING

Dialog summarization has its unique challenges compared to other general summarization tasks as it requires wide-coverage natural language understanding of in-domain knowledge as well as semantic relevance between the speaker and the sentences. We develop seven self-supervised pre-training strategies to enhance BERT for the dialog summarization task and potentially other downstream tasks. To leverage the useful knowledge from the pre-trained BERT, we do not train a new version of BERT from scratch, but rather start from the pre-trained BERT-base checkpoint and use our pre-training methods to enhance the model's semantic and structural understanding of dialogs, which will be transferred and benefit on the summarization task.

Pre-processing: Adopting a similar approach as in Liu and Lapata [17], we pre-process the datasets by adding the "[CLS]" token at the start and the "[SEP]" token at the end of each sentence. The "[CLS]" token is used as the sentence representation in the pre-training tasks as well as downstream extractive summarization task. Let t_i denote the vector of the i-th "[CLS]" token from the top layer, then t_i can be used as the i-th sentence's representation. Then we build an extractive summarization model on top of this encoder by stacking several inter-sentence Transformer layers to capture the dialog level features for extracting sentences. Finally, a linear layer is applied to predict whether to choose the sentence to form the summary.

Sentence mask: The original BERT model has two self-supervised pre-training objectives: Masked language modeling (MLM) and next

sentence prediction (NSP) [13]. In the MLM task, a small percentage of the input subword tokens are randomly masked, and the training objective is to predict the original token. For our use case, we modify this subword masking to sentence masking as our down-stream task is to select sentences for summarization. We first mask some sentences within a dialog with probability P_m^{sen} and put these masked sentences into a candidate pool T^m. The objective is to predict the correct sentence from the pool for each masked sentence i within the dialog. We replace the selected sentence i with a special token "[UNK]" and extract the embedding vector of the corresponding "[CLS]" token, D_i, as the sentence embedding. Then, we concatenate all candidate sentences and use the same model to obtain the sentence embedding S^m for each sentence in the candidate pool T^m to calculate the cosine similarity between sentence i and each candidate sentence j in T^m:

$$\theta(i, j) = cos(D_i, S_j^m) \tag{1}$$

We use a ranking loss to maximize the margin between the reference sentence and other sentences:

$$l_m = max\{0, \gamma - \theta(i, j) + \theta(i, k)\} \tag{2}$$

where γ is a tunable hyper-parameter, i refers to the masked sentence, j refers to the reference sentence in the candidate pool T^m, while k refers to the rest of sentences in T^m.

Speaker mask: Speaker mask can be considered as a special case of the MLM task in BERT, but instead of masking input subword tokens, we are masking the special speaker tokens. In our experiments with the TWEETSUMM dataset, we use "CustomerStart" and "AgentStart" as the special speaker tokens (see Table 1 for an illustration of the input format). During the self-training, we mask the speaker tokens with probability P_m^{spk} using the special token "[MASK]" and the task is to predict which speaker the selected speaker token is from.

Sentence replacement: The *sentence replacement* task is to select sentences with probability P_r^{sen} in a dialog and replace them with sentences selected at random from the entire corpus. The model is trained to predict whether the sentence is replaced.

Sentence switch: The *sentence switch* task is similar with the *sentence replacement* task except for that the sentence is replaced with another sentence from the same dialog. In other words, each sentence has a probability of P_s^{sen} to be put in another position within the same dialog and the object is to predict whether the sentence is in its original position or not.

Speaker switch: Similar to the *sentence switch* task, we switch the special speaker tokens, i.e., "CustomerStart" and "AgentStart". We randomly select speaker tokens with probability P_s^{spk} and switch it with its counterpart. The model then predicts whether the speaker is switched or not.

Sentence insertion: For this task, we randomly select the insertion positions within a dialog with a pre-defined probability P_{ins}^{sen} and insert sentences randomly selected from the entire corpus into these positions. This task aims to predict whether each sentence is from the original dialog or not after the insertion.

Mixed strategy: We combine the *sentence replacement*, *sentence switch*, and *sentence insertion* in this task where each dialog has an equal probability to get one of the three operations (therefore 33.33% probability to get each operation). And the task is to predict whether each sentence is corrupted or not.

Except for the *sentence mask* and the *speaker mask* tasks, binary cross-entropy loss is used for each of the other five tasks. Through learning each of the pre-training tasks presented above, the model is expected to learn contextualized embedding with in-domain knowledge, which will then be used for the down-stream summarization task. The *sentence mask*, *sentence replacement*, *sentence switch*, and *sentence insertion* tasks aim to improve the model's semantic understanding of the sentence in the context of the entire dialog while the *speaker mask* and *speaker switch* tasks aim to enhance the model's ability to capture different roles from the dialog.

4 DIALOG SUMMARIZATION WITH TEACHER-STUDENT TRAINING AND CONFIDENCE WEIGHTS

Since extractive summary annotation is only available for a small subset of a large corpus of in-domain dialog data, the goal is to use a teacher-student training framework to make use of both the small annotated dataset for training and validation, as well as the large unlabeled data. A fully-supervised model (teacher) is first trained on the few-shot annotated data and used to generate pseudo labels for the large scale unlabeled in-domain dataset. We propose an approach to improve teacher-student training with confidence weights. Re-weighting noisy labels with different weighting schemes has been explored in previous work including meta learning and rationale extraction [3, 24]. We propose a definition of confidence weights specifically for our extractive summarization task. Let P_{ij} denote the probability of sentence i being selected for summary for dialog j estimated by the teacher model, and $Q_{ij} \in \{0, 1\}$ denote if sentence i is predicted as summary for dialog j. We choose the sentences with top 4 predicted probabilities in one dialog as its summary. So Q_{ij} are defined as

$$Q_{ij} = \begin{cases} 1 & \text{if } P_{ij} \text{ is in top 4 among } P_{\cdot j} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Then we define the confidence weight of each sentence, C_{ij}, and each dialog, C_j, as

$$C_{ij} = P_{ij}Q_{ij} + (1 - P_{ij})(1 - Q_{ij}) \tag{4}$$

$$C_j = \sum_{i=1}^{n_j} (C_{ij})/n_j \tag{5}$$

C_{ij} reflects how confident the teacher model is about the predicted label. And the confidence weight for an entire dialog C_j is calculated by the mean of C_{ij}s for individual sentences in this dialog. The pseudo-labeled data is sampled based on their confidence weights and augmented with the original labeled data to be used to train a student model. This weighted augmentation explicitly accounts for the teacher's confidence on the generated pseudo-labels with re-weighting to focus more on the pseudo-labeled samples that the

Table 1: An example of a customer agent dialog and its extractive summary

CustomerStart	My watchlist is not updating with new episodes.
CustomerStart	Any idea why?
AgentStart	Apologies for the trouble, Norlene! We're looking into this.
AgentStart	In the meantime, try navigating to the episode manually.
CustomerStart	Tried logging out/back in, that didn't help
AgentStart	Sorry!
AgentStart	We assure you that our team is working hard to investigate, and we hope to have a fix ready soon!
CustomerStart	Thank you!
CustomerStart	Some shows updated overnight, but others did not...
AgentStart	We definitely understand, Norlene.
AgentStart	For now, we recommend checking the show page as the new episodes will be there
CustomerStart	As of this morning, the problem seems to be resolved.
CustomerStart	Watchlist updated overnight with all new episodes.
CustomerStart	Thank you for your attention to this matter!
CustomerStart	I love Hulu
AgentStart	Awesome!
AgentStart	That's what we love to hear.
AgentStart	If you need anything else, we'll be here to support!
Extractive summary	
CustomerStart	My watchlist is not updating with new episodes.
AgentStart	We assure you that our team is working hard to investigate, and we hope to have a fix ready soon!
AgentStart	For now, we recommend checking the show page as the new episodes will be there
AgentStart	If you need anything else, we'll be here to support!

teacher is more confident on compared to the less certain ones during teacher-student training. The teacher-student training schedules are repeated till a convergence criterion is satisfied. At each iteration, the model parameters learnt from the previous iteration are used to initialize the current teacher model before re-training the model on the augmented data to learn the new parameters.

5 EXPERIMENTS AND EVALUATION

To demonstrate the advantages of our proposed pre-training methods and teacher-student learning in applications with limited labels, we conduct various experiments on the recently released TWEETSUMM dataset [6] for the real-world use case of customer service dialog summarization. In this section we describe the experimental setup and discuss implementation and evaluation details.

5.1 TWEETSUMM Dataset

In the real-world scenario of customer service chat, a customer contacts a support center to ask for help or raise complaints and a human agent tries to solve the issues. A short summary at the end of each conversation can help other agents quickly grasp the core content of a dialog without needing to review the entire transcripts. Providing automatic dialog summaries has thus become one of the most important features in a customer service system.

Traditionally, public data and resources on customer service dialogs were very scarce due to customer data privacy concerns and intellectual property protection. However, this situation changed when the *Customer Support On Twitter* dataset was made available on Kaggle[1]. It is a large scale dataset based on conversations between customers and customer support agents on *Twitter.com* [11].

[1]https://www.kaggle.com/datasets/thoughtvector/customer-support-on-twitter

Table 2: Dataset Descriptive Statistics

	Overall	Customer	Agent
Customer Support on Twitter Dataset (70,191 dialogs)			
Vocabulary size	285337	234222	94150
Avg # turns per dialog	8.0	4.3	3.7
Avg # sentences per turn	2.2	1.9	2.5
Avg # words per sentence	9.5	10.5	8.7
TWEETSUMM Dataset (1,100 dialogs)			
Vocabulary size	21759	15981	10515
Avg # turns per dialog	10.2	5.5	4.7
Avg # sentences per turn	2.1	1.8	2.4
Avg # words per sentence	9.7	10.4	9.0
Extractive Summary Dataset (3,056 summaries)			
Vocabulary size	13982	9971	7211
Min # sentences per summary	2	1	1
Max # sentences per summary	12	8	8
Avg # sentences per summary	4.3	2.1	2.2
Avg # words per sentence	15.6	16.0	15.2

This dataset is the first publicly available dataset for real-world customer support which consists of 3 million tweets from 20 big companies such as Amazon, Apple, Uber, Delta, Spotify, etc. Feigenblat et al. [6] randomly sampled 1,100 dialogs to be manually annotated for summaries to construct and release the TWEETSUMM dataset[2]. Annotators were instructed to highlight the most salient sentences in the dialog for extractive summarization labeling, and 3 annotations from 3 annotators were collected per dialog, therefore about 3300 extractive summaries were collected to form the extractive dataset. We discard dialogs with less than 6 or greater than 20 turns. Then we add "AgentStart" or "CustomerStart" at the beginning of each turn as a mark of speaker. Since the dialogs are extracted from twitter threads, there are many tweet links in the data. We replace all the links with a specific token "TweetLinkToken". In our final dataset, we have a total of 3056 dialog and extractive summary pairs. Table 1 presents an example of a customer agent dialog and its extractive summary and Table 2 presents the descriptive statistics of the dataset. We split the 3,056 dialogs and their associated summaries into approximately 80% for training, 10% for validation, and 10% for testing.

5.2 Baselines

We compare our methods with the following two baselines:

LEAD-4. Feigenblat et al. [6] did an analysis on the positions of the sentences selected for extractive summaries in the labeled dataset and found that in 85% of dialogs sentences from the first customer turn were selected and in 52% of the dialogs sentences from the first agent turn were selected. This pattern corroborates with a typical customer service dialog which starts with a customer describing an issue or asking a question, followed by several conversation rounds for more context, and ends with the agent

[2]https://github.com/guyfe/Tweetsumm

taking actions to resolve the issue or escalating to another channel. Therefore, the LEAD-4 model which selects the first two sentences from the customer side and the first two sentences from the agent side should be considered a competitive baseline, especially for summarizing customer issues.

PreSumm. The PreSumm model [17] is similar to our approach where an encoder creates sentence representations, upon which a classifier layer predicts a binary label for each sentence indicating whether it is selected to form the summary. While PreSumm uses vanilla BERT as the sentence encoder we use an enhanced BERT model which is obtained by pre-training using self-supervised tasks that are suitable for dialog summarization.

5.3 Implementation and Training Details

The training process consists of two phases. First, we use the pre-training tasks to enhance BERT. All our experiments start from the publicly available 'bert-base-uncased' version. After pre-training we further fine-tune the model for summarization on the TWEETSUMM dataset.

For training we use the AdamW optimizer[19] with a batch size of 16, an input sequence length of 512, dropout rate of 0.1 and a learning rate from 1e-6 to 1e-7. For self-supervised pre-training, we train the model until the train loss converges or reaches the maximum epoch of 20. We fine-tune the resulting model for summarization task until the validation loss converges or reaches the maximum training epoch of 20. We set P_m^{sen}, P_m^{spk}, P_r^{sen}, P_s^{sen}, P_s^{spk}, and P_{ins}^{sen} to 0.5.

When predicting summaries for a new dialog, we use the model to obtain the scores for each sentence and use the top 4 sentences with the highest scores to form the summary. The choice of top 4 sentences is based on the descriptive statistics reported in Table 2 that the average length of the ground truth extractive summaries in TWEETSUMM is 4 sentences. During sentence selection we use Trigram Blocking [22] to reduce redundancy where a candidate sentence is skipped if there already exists a sentence in the summary that trigram overlaps with this candidate. Trigram Blocking ensures that no two similar sentences are selected for summarization at the same time to maximize diversity in the summary.

6 RESULTS

We evaluate the quality of summarization using ROUGE [16]. We report unigram and bigram overlap (ROUGE-1 and ROUGE-2) as a means of assessing informativeness and the longest common sub-sequence (ROUGE-L) as a means of assessing fluency.

The results from the proposed self-supervised pre-training methods and teacher-student learning are summarized in Table 3. The first block in the table includes the LEAD-4 baseline and the PreSumm baseline. The second block includes extractive models fine-tuned for summarization on the TWEETSUMM dataset after applying various self-supervised pre-training strategies to enhance BERT. The third block in the table presents the performance of semi-supervised teacher-student learning (TSL), and the fourth block presents the results where both the enhanced BERT and teacher-student learning are applied. All of our proposed methods have consistently outperformed baselines for all settings by up to 3.91% in R-1, 5.07% in R-2, and 4.06% in R-L when comparing to the PreSumm model. The best performance is achieved by the Mixed strategy + TSL model across all ROUGE scores, demonstrating that both self-supervised pre-training on in-domain dataset and teacher-student training to leverage large-scale unlabeled data have an incremental positive impact on the summarization task performance.

Table 3: ROUGE F-1 Score results on TWEETSUMM test set

Method	R-1(%)	R-2(%)	R-L(%)
Baselines			
LEAD-4	53.43	42.78	52.81
PreSumm	66.07	57.91	65.45
Self-supervised Pre-training (SSPT)			
Sentence mask	67.34	59.55	66.82
Speaker mask	66.11	57.88	65.46
Sentence replacement	69.01	61.91	68.57
Sentence switch	68.91	61.70	68.42
Speaker switch	68.43	60.98	67.94
Sentence insertion	68.93	61.64	68.46
Mixed strategy	69.54	62.67	69.08
Semi-supervised Teacher-Student Learning (TSL)			
TSL	67.38	59.45	66.87
SSPT + TSL			
Sentence mask + TSL	68.35	60.60	67.84
Speaker mask + TSL	68.04	60.56	67.51
Sentence replacement + TSL	69.32	62.26	68.87
Sentence switch + TSL	68.84	61.56	68.31
Speaker switch + TSL	68.17	60.79	67.72
Sentence insertion + TSL	69.11	61.82	68.65
Mixed strategy + TSL	**69.97**	**62.98**	**69.51**

6.1 Self-supervised Pre-training (SSPT) Results

On the other hand, not all self-supervised pre-training tasks yield equal results. The sentence level tasks perform better than the speaker level tasks. This could be due to the fact that the sentence level tasks are better at helping the model understand the dialog context, and the speaker level tasks are too easy for the pre-training models as their accuracy reaches 99% quickly. Furthermore, masking methods (both sentence mask and speaker mask) show the least favorable results compared to other methods. This is because the task of masking methods is different from the downstream summarization task with a binary cross-entropy loss. One of the main benefits of self-supervised pre-training is faster convergence. We plot the ROUGE scores on the development set during the fine-tuning phase in Figure 1, which demonstrates that not only do the summarization models built upon the enhanced BERT yield superior results, they also converge faster than that built upon the original BERT. Specifically, the baseline PreSumm model takes 15 epochs to converge, the speaker mask model takes 12 epochs to converge, while all other pre-training tasks take less than 10

epochs to converge during the summarization fine-tuning phase. For clarity, we only plot ROUGE-1 F1 scores in this figure and include the plots for ROUGE-2 and ROUGE-L in the Appendix, which have shown a very similar trend. This demonstrates that a good pre-trained encoder could dramatically save time and resources for downstream tasks.

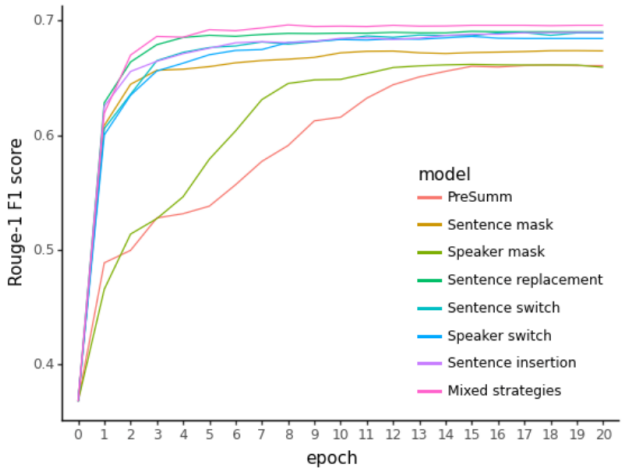

Figure 1: ROUGE-1 during the summarization fine-tuning phase

6.2 SSPT + TSL

The results in Table 3 also demonstrate how our proposed self-supervised pre-training on in-domain dataset and teacher-student learning strategies compliment each other. When pre-training does not yield significant improvement, the teacher-student learning would be especially beneficial where useful information is gained from the student models. On the other hand, if large performance improvements are achieved during pre-training, then the additional benefit of additional teacher-student training is lower. For example, compared to the PreSumm baseline, adding pre-training with speaker mask task only yields 0.01% increase in R-L (from 65.45% to 65.46%), while teacher-student learning yields an additional 2.05% increase (from 65.46% to 67.51%). On the contrary, pre-training with mixed strategy brings a significant 3.63% gain in R-L from the PreSumm model (from 65.45% to 69.08%), but additional teacher-student learning has yielded merely another 0.43% gain in R-L. This shows that in practice when there is enough computing resources, there are advantages to apply both strategies to ensure optimal performance.

6.3 SSPT vs TSL in low-resource data settings

In reality, most summarization settings have the challenge of limited labels due to the fact that high-quality annotated data for summarization is both time consuming and expensive to obtain. Hence we now compare self-supervised pre-training vs semi-supervised teacher-student learning in different low-resource data setups with limited labels. We use the performance on the downstream summarization task for this comparison. First, we investigate the effect

Table 4: SSPT results in the very low- (50 labels) and low- (500 labels) resource settings

Method	R-1(%)	R-2(%)	R-L(%)
50 labels			
PreSumm	48.13	36.89	47.46
Sentence mask	56.27	46.07	55.61
Speaker mask	49.26	37.45	48.58
Sentence replacement	60.16	51.15	59.52
Sentence switch	57.38	46.76	56.53
Speaker switch	57.01	46.55	56.20
Sentence insertion	60.87	50.44	59.97
Mixed strategy	59.64	50.16	58.90
500 labels			
PreSumm	63.73	54.63	63.02
Sentence mask	67.19	59.16	66.60
Speaker mask	65.44	57.07	64.81
Sentence replacement	66.10	57.75	65.51
Sentence switch	67.30	59.37	66.70
Speaker switch	66.53	58.53	65.99
Sentence insertion	67.82	60.26	67.26
Mixed strategy	69.37	62.10	68.90

of various pre-training objectives in both very low- (50 labels) and low (500 labels) resource settings and the results are presented in Table 4. We notice that the magnitude of performance gain is much more significant with 50 labels compared to 500 labels or the full 3056 labels as presented in Table 3. Specifically, the magnitude of performance gain after pre-training with the mixed strategy compared to the PreSumm baseline is 11.51% in R-1, 13.27% in R-2 and 11.44% in R-L with 50 labels, while 5.64% in R-1, 7.47% in R-2, 5.88% in R-L with 500 labels, and 3.47% in R-1, 4.76% in R-2 and 3.63% in R-L with the full 3056 labels. This shows that the benefit of self-supervised pre-training is especially pronounced when labels are scarce. And even more impressively, the mixed strategy model using only 500 labels has already outperformed the baseline PreSumm model which is directly fine-tuned on the entire 3056 labels without any no pre-training, where the former achieves 69.37% in R-1, 62.10% in R-2, and 68.90% in R-L while the latter yields 66.07% in R-1, 57.91% in R-2, and 65.45% in R-L. Furthermore, the ROUGE scores achieved by the mixed strategy model using 500 labels are comparable to the ROUGE scores from the mixed strategy model using the full 3056 labels as presented in Table 3, further demonstrating that self-supervised pre-training is an extremely effective technique when labels are scarce.

Next, we experiment with different labeled and unlabeled summarization data sizes for teacher-student learning. In Table 5, we show the results of ROUGE evaluated on the annotated test set for different models. In the very low resource setup where there are only 50 labels, we notice that adding more unlabeled data to train the student model leads to very small gains, while when there are 500 labels, adding more unlabeled data proves to be more effective. This is expected as the majority of information is retrieved from

Table 5: TSL results in different labeled and unlabeled data sizes

labeled:unlabeled	R-1(%)	R-2(%)	R-L(%)
50 labels			
50:50	48.51	37.20	47.76
50:500	49.04	37.65	48.24
50:5000	49.12	37.56	49.29
50:50000	49.31	37.69	49.42
500 labels			
500:50	51.24	39.75	50.37
500:500	58.96	48.32	58.19
500:5000	66.04	57.96	65.41
500:50000	67.77	59.45	67.14

labels therefore semi-supervised teacher-student learning can only be effective when there is a reasonable amount of labeled data.

When we compare the results for 50 labels between Table 4 and Table 5, as well as 500 labels between these two tables, it is clear that given the same amount of labels, pre-training techniques outperform the teacher-student learning technique and the differences between the two are more significant when the available labeled data are extremely limited. This observation indicates a more efficient knowledge transfer for in-domain self-supervised pre-training compared to teacher-student learning. Therefore, under computational or time constraints, we recommend choosing pre-training with a mixed strategy as a lightweight technique, as it has demonstrated a clear advantage and benefits especially when annotated summarization labels are scarce.

7 CONCLUSION AND FUTURE WORK

In this paper we showed the benefits of seven self-supervised methods to enhance an off-the-shelf pre-trained BERT (sentence mask, speaker mask, sentence replacement, sentence switch, speaker switch, sentence insertion, and mixed strategy) as well as a teacher-student framework for extractive dialog summarization with limited labels. The self-supervised pre-training process helps the model to capture a better semantic relevance between speaker and the sentences compared to other BERT models trained on documents, which can be crucial for the downstream summarization task to preserve the most important information in the whole dialog. The teacher-student learning helps to make use of the unlabeled data to address the shortcomings of lack of large scale labels. Our experiments on the TWEETSUMM corpus indicate that our a combination of self-supervised pre-training and teacher-student learning leads to substantial improvement in ROUGE scores compared to direct fine-tuning. The resulting models demonstrate clear superior performance and converge faster when learning on the summarization task. The best performance is achieved by mixing multiple pre-training tasks combined with Teacher Student Learning model across all ROUGE scores, demonstrating that both self-supervised pre-training on in-domain dataset and teacher-student training to leverage large-scale unlabeled data have an incremental positive impact on the summarization task performance. Therefore in practice when there is enough computing resources, there are clear advantages to apply both strategies to ensure optimal performance. Furthermore, in our investigation in different data resource settings, we found that pre-training is more effective than teacher-student learning, especially in low resource settings.

We believe there is still much room for future improvements and our work will help foster further research in the real world scenario of summarizing for goal oriented dialog texts. In this work we focused on extractive summarization and leave abstractive summarization to future work. Previous work has shown that the combination of extractive and abstractive objectives can help generate better summaries [8]. An interesting direction is to explore a two-stage approach where the extractive and the abstractive stages share the same encoder. In our experiments, we only used the 'bert-base-uncased' model. It will inspire more investigation into other language models such as GPT2, Roberta, and T5. It will also be worthwhile to explore whether starting pre-training with a different version of bert model, such as Conversation-bert leads to faster convergence. In this paper, we have shown that not all pre-training tasks are equal and our pre-training strategy with a mixture of different tasks produced the best performance. In future works, we can explore the direction of reinforcement learning where we can learn a policy to decide which pre-training task to perform at each step that maximizes the "reward function" which is the downstream summarization performance.

REFERENCES

[1]]ConversationalBert [n. d.]. Conversational-BERT. https://huggingface.co/DeepPavlov/bert-base-cased-conversational. Accessed: 2022-10-30.

[2] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[3] Meghana Moorthy Bhat, Alessandro Sordoni, and Subhabrata Mukherjee. 2021. Self-training with few-shot rationalization: Teacher explanations aid student in few-shot nlu. *arXiv preprint arXiv:2109.08259* (2021).

[4] Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 29.

[5] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555* (2020).

[6] Guy Feigenblat, Chulaka Gunasekara, Benjamin Sznajder, Sachindra Joshi, David Konopnicki, and Ranit Aharonov. 2021. TWEETSUMM - A Dialog Summarization Dataset for Customer Service. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 245–260. https://doi.org/10.18653/v1/2021.findings-emnlp.24

[7] Xiachong Feng, Xiaocheng Feng, Bing Qin, and Xinwei Geng. 2020. Dialogue discourse-aware graph model and data augmentation for meeting summarization. *arXiv preprint arXiv:2012.03502* (2020).

[8] Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-Up Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 4098–4109. https://doi.org/10.18653/v1/D18-1443

[9] Jia-Chen Gu, Chongyang Tao, Zhen-Hua Ling, Can Xu, Xiubo Geng, and Daxin Jiang. 2021. MPC-BERT: A pre-trained language model for multi-party conversation understanding. *arXiv preprint arXiv:2106.01541* (2021).

[10] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 8342–8360. https://doi.org/10.18653/v1/2020.acl-main.740

[11] Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2018. Towards automated customer support. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*. Springer, 48–59.

[12] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*

(2019).
[13] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*. 4171–4186.
[14] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451* (2020).
[15] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).
[16] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013
[17] Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3730–3740. https://doi.org/10.18653/v1/D19-1387
[18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
[19] Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam. (2018).
[20] Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020. XtremeDistil: Multi-stage Distillation for Massive Multilingual Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2221–2234. https://doi.org/10.18653/v1/2020.acl-main.202
[21] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*.
[22] Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304* (2017).
[23] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
[24] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to Reweight Examples for Robust Deep Learning. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 4334–4343. https://proceedings.mlr.press/v80/ren18a.html
[25] Hong Wang, Xin Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Self-supervised learning for contextualized extractive summarization. *arXiv preprint arXiv:1906.04466* (2019).
[26] Xiaolong Wang and Abhinav Gupta. 2015. Unsupervised Learning of Visual Representations Using Videos. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 2794–2802. https://doi.org/10.1109/ICCV.2015.320
[27] Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. 2020. Adaptive self-training for few-shot neural sequence labeling. *arXiv preprint arXiv:2010.03680* (2020).
[28] Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv preprint arXiv:1905.06566* (2019).
[29] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. *arXiv preprint arXiv:1807.02305* (2018).
[30] Yingying Zhuang, Yichao Lu, and Simi Wang. 2021. Weakly Supervised Extractive Summarization with Attention. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, Singapore and Online, 520–529. https://aclanthology.org/2021.sigdial-1.54

A SUPPLEMENT PLOTS TO *FIGURE 1*

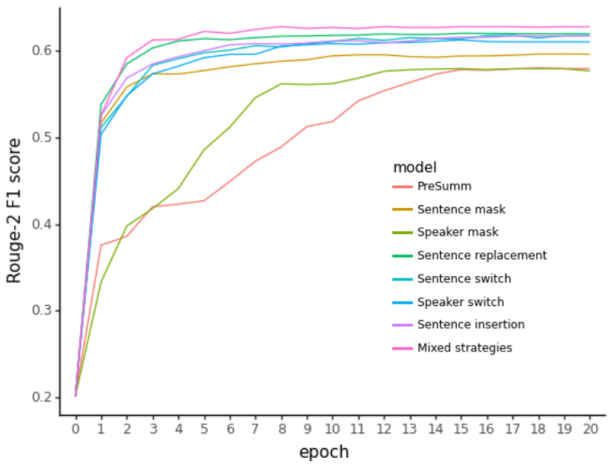

Figure 2: ROUGE-2 during the summarization fine-tuning phase

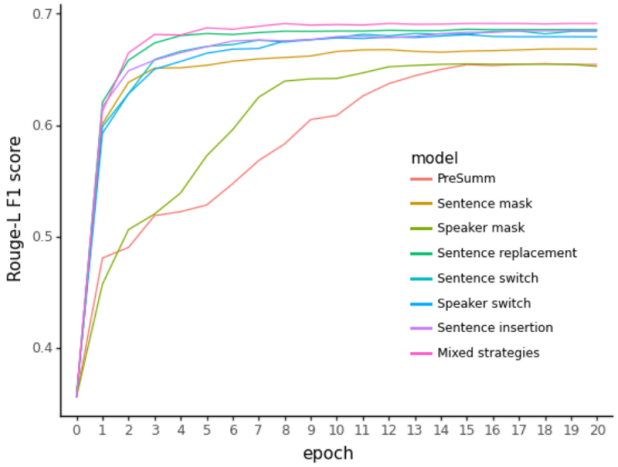

Figure 3: ROUGE-L during the summarization fine-tuning phase