Noise-injected Consistency Training and Entropy-constrained Pseudo Labeling for Semi-supervised Extractive Summarization

Yiming Wang[1,2,3,†], **Qianren Mao**[1,2,†], **Junnan Liu**[1,2], **Weifeng Jiang**[1,2], **Hongdong Zhu**[1,2], **Jianxin Li**[1,2,*]

[1] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, China.
[2] The State Key Laboratory of Software Development Environment, Beihang University, China.
[3] Institute of Artificial Intelligence, Beihang University, China.
{wangym,maoqr,liujn,jiangwf,zhuhd,lijx}@act.buaa.edu.cn

Abstract

Labeling large amounts of extractive summarization data is often prohibitive expensive due to time, financial, and expertise constraints, which poses great challenges to incorporating summarization system in practical applications. This limitation can be overcome by semi-supervised approaches: *consistency-training* and *pseudo-labeling* to make full use of unlabeled data. Researches on the two, however, are conducted independently, and very few works try to connect them. In this paper, we first use the noise-injected consistency training paradigm to regularize model predictions. Subsequently, we propose a novel entropy-constrained pseudo labeling strategy to obtain high-confidence labels from unlabeled predictions, which can obtain high-confidence labels from unlabeled predictions by comparing the entropy of supervised and unsupervised predictions. By combining consistency training and pseudo-labeling, this framework enforce a low-density separation between classes, which decently improves the performance of supervised learning over an insufficient labeled extractive summarization dataset.

1 Introduction

Text summarization is a challenging task that generates a condensed version of an input text that captures the original's core meaning. In this paper, we focus on extractive summarization since it usually generates semantically and grammatically correct sentences (Liu and Lapata, 2019; Zhong et al., 2019a; Zhou et al., 2020; Zhong et al., 2020). The extractive summarization typically requires to label each sentence in massive documents for model training. However, acquiring well-annotated labels is a costly process, and labeling every sentence would be labor-intensive and error-prone due to subjective judgments of human. This motivates research on Semi-Supervised Learning (SSL) methods which focus on how to effectively utilizes abundant unlabeled data, to further improve extractive summarization performances.

Towards this goal, we first revisit an effective semi-supervised method, consistency training (Xie et al., 2020a). The consistency training leverages voluminous unlabeled data and employs advanced data augmentation methods to generate diverse and realistic noisy source text, forcing the model to be consistent with these noises. The consistency training has been extensively applied on the classification problems, such as Text Classification (Xie et al., 2020a; Liu et al., 2021a), Image Recognition (Laine and Aila, 2017; Tarvainen and Valpola, 2017; Miyato et al., 2019; Verma et al., 2019; Xie et al., 2020b). However, how does the consistency training work on semi-supervised extractive summarization tasks is still unclear.

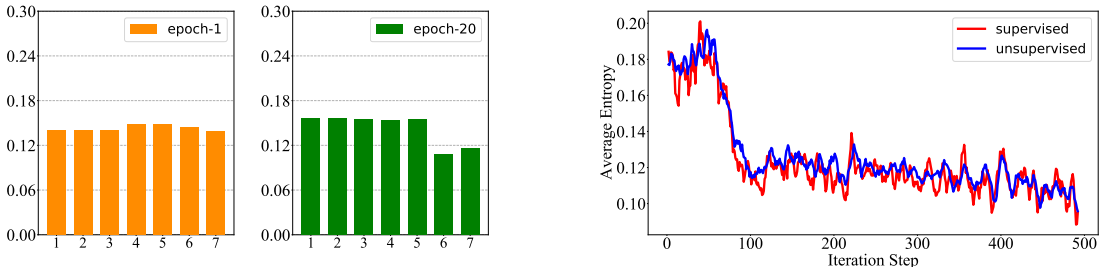
We investigate the noise-injected consistency training for semi-supervised extractive summarization to encourage a consistent reason of model decision (summary and non-summary) under data perturbation. This framework makes sense intuitively because a good supervised model should be robust to any slight change in an input example. Namely, encouraging local change by injecting a slight noise in a diverse perturbation manner can improve the summarization effectiveness.

Nevertheless, the consistency regularized semi-supervised framework usually suffers from insufficient supervision. When labeled data is limited, the model is easy to over-fitting. Extensive unlabeled data will then make the model suffer from the gradual drift problem and impede further improvements of the model. To address this problem, we develop new methods of selection and exploitation for pseudo labels to explore all unlabeled samples for the semi-supervised summarization cycle.

Prior pseudo labeling work (Lee et al., 2013; Sohn et al., 2020; Rizve et al., 2021) mainly fo-

[†] These two authors contributed equally.
[*] Jianxin Li is the corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 6447–6456
October 12–17, 2022.



(a) Prediction probability distributions

(b) Average entropy changing curve

Figure 1: (a) Prediction probability distributions of the same sample in different epochs before and after convergence. (b)The average entropy changing curve of labeled and unlabeled samples before convergence during training.

cuses on the confidence of the predictions to pick up the class used as if they are accurate labels. However, the summarization task exposes potential problems with this approach. The confidence of the prediction results is hard to change significantly along with model convergence (e.g., using the BERT-based model). As shown in the second sub-figure in Fig. 1(a), almost all prediction probability of sentences has achieved an undifferentiated score between 0.12 to 0.16 in the convergence epoch, which means almost all sentences are mapped into a small area for classification. Therefore, it is challenging to set a fixed threshold to determine a proper confidence score, and it is easy to produce low-quality pseudo labels. We divert attention to entropy − a metric for measuring uncertainty, to address this problem. As is shown in Fig 1(b), the prediction entropy of supervised and unsupervised data are comparable, which enables us to constrain the unsupervised entropy with supervised entropy without any external threshold value. We introduce entropy-constrained pseudo labeling to avoid fixed threshold adoption, which ensures the entropy of prediction results of unlabeled data adjust that of labeled samples adaptively.

The final framework is **CPSUM**[1], which combines the Noise-injected **C**onsistency training with the Entropy-constrained **P**seudo labeling for Semi-supervised Extractive **SUM**marization. Experimental results demonstrate that our approach achieves the state-of-the-art in low-resource scenarios with 10, 100, and 1000 labeled examples on the target corpus. The main contributions of our method are:

- To the best of our knowledge, this is the first work to explore the feasibility of consistency-training and pseudo-labeling for semi-supervised extractive summarization tasks.
- Proposing a novel threshold-free approach selecting reliable pseudo-labels with the average entropy comparison, which is well-adapted to extractive summarization tasks.
- Extensive evaluations demonstrate that consistency training and pseudo-labeling with unsupervised corpus could greatly improve the performance of the text summarization model on a limited dataset.

2 Related Work

2.1 Extractive Summarization

Extractive summarization selects the most representative sentences within a document and subsequently splices them into the final summary. Approaches for it are constantly updated. With classical networks, RNN-based (Nallapati et al., 2017; Zhou et al., 2018), Transformer-based (Zhong et al., 2019b; Liu et al., 2021b) are adopted. Pre-trained summarization models have achieved great success, such as the notable BERTSUMEXT (Liu and Lapata, 2019; Liu, 2019) which is the first work to use the BERT (Devlin et al., 2019) for extractive summarization. However, current extractive summarization models still heavily rely on many parallel data to achieve salient performance. Little work has focused on low-resourced settings where handcrafted labels for sentences are limited or even unavailable. To fill this gap, in this work, we introduce a novel semi-supervised framework to alleviate the dependence on labeled summaries.

2.2 Consistency Regularization

In recent work, consistency regularization methods for semi-supervised learning (Bachman et al., 2014) have been shown to work well on many classification tasks (Xie et al., 2020a; Liu et al., 2021a).

[1]Code and data available at: https://github.com/OpenSUM/CPSUM.

Their work can match and even outperform purely supervised learning that uses affluent labeled data.

The consistency training methods regularize model predictions invariant to noise applied to unlabeled examples. Tarvainen and Valpola (2017) prove that a model trained with noisy labeled data learns to give consistent predictions around labeled data points. Additionally, advanced data augmentation methods (Xie et al., 2020a) can improve consistency training performance effectively.

2.3 Pseudo-labeling

Pseudo-labeling (Lee et al., 2013) is an efficient semi-supervised learning method by generating pseudo-labels to expand labeled data. For selecting reliable pseudo-labels, FixMatch (Sohn et al., 2020) creates a selection criterion based on the confidence threshold. After that, considering poor network calibration, UPS (Rizve et al., 2021) introduces model uncertainty criterion based on prediction result confidence. U^{2}PL (Wang et al., 2022) does the opposite and fully considers the value of some unreliable pseudo-labels. However, the performance of pseudo-labels on the extractive summarization task remains to be evaluated, and the methods above are not suitable for this task directly.

3 Proposed Framework

The overview illustration of our framework is shown in Fig. 2. It is composed of two components: noise-injected consistency training and entropy-constrained pseudo labeling. Specific to extractive summarization, we use the base version of BERT (Devlin et al., 2019) to implement our models in all experiments. We give detail description of two components in Sec.3.1 and Sec.3.2.

3.1 Noise-injected Consistency Training

The consistent regularization of semi-supervised learning leverages unlabeled data, employs data augmentation methods to inject noisy data, and enforces the summarization model by encouraging consistent predictions.

Data Augmentation. The unlabeled noise examples, specifically those produced by advanced data augmentation methods, have been proved to be crucial for consistency training (Xie et al., 2020a). Our augmentation method refers to Jiao et al. (2020). We replace single-piece words (Wu et al., 2019) by predictions of the BERT masked language model and retrieve the most similar words as word replacements for multiple-pieces words by using the word embedding in GLOVE (Pennington et al., 2014).

Consistency Training. The robust summarization model should also be invariant for documents with similar content. Hence, we leverage consistency learning to regularize model predictions to be invariant to slight noise applied to input examples.

The inputs of the framework are labeled texts x, unlabeled texts $x^{'}$, and noise injected unlabeled texts $x^{''}$. We use y^* to denote the gold summaries of labeled texts. Then we use f_θ to represent the distributions of model predictions, where θ refers to the model's parameters. Firstly, we feed the labeled text x into the model to get the predictions and calculate the supervised loss:

$$\mathcal{L}_l = \frac{1}{|X_l|} \sum_{x \in X_l} l(y^*, f_\theta(y|x)), \quad (1)$$

where X_l is a set containing $|X_l|$ labeled data x. We then generate a noised version $x^{''}$ of the unlabeled text $x^{'}$ using the aforementioned data augmentation method. Both unlabeled texts and noised unlabeled texts are fed to the summarization model, and then we get the output distribution of original unlabeled data $f_{\tilde{\theta}}(y^{'}|x^{'})$ and the additional noised version of augmented unlabeled data $f_\theta(y^{''}|x^{''})$. We then calculate the unsupervised loss between unlabeled texts and augmented unlabeled texts:

$$\mathcal{L}_u = \frac{1}{|X_u|} \sum_{(x',x'') \in X_u} l(f_{\tilde{\theta}}(y^{'}|x^{'}), f_\theta(y^{''}|x^{''})), \quad (2)$$

where X_u is a set of pairs containing $|X_u|$ unlabeled data x' and the corresponding augmented data x'', and $\tilde{\theta}$ is just a copy of the current parameters θ indicating that the back-propagation of the gradient is truncated. We use KL divergence loss to perform consistency training.

Finally, we combine supervised cross-entropy loss and supervised consistency loss, and train the model by minimizing the combined loss: $\mathcal{L}_f = \mathcal{L}_l + w(t)\mathcal{L}_u$, where $w(t)$ is the ramp-up weight balancing supervised and unsupervised learning.

3.2 Entropy-constrained Pseudo Labeling

Generally, the prediction results with the highest predicted probability of unlabeled data could be adopted as pseudo labels. However, low-quality pseudo labels may harm model training.

To overcome this problem, we introduce a method named entropy-constrained pseudo labeling to select reliable pseudo labels. We argue that if

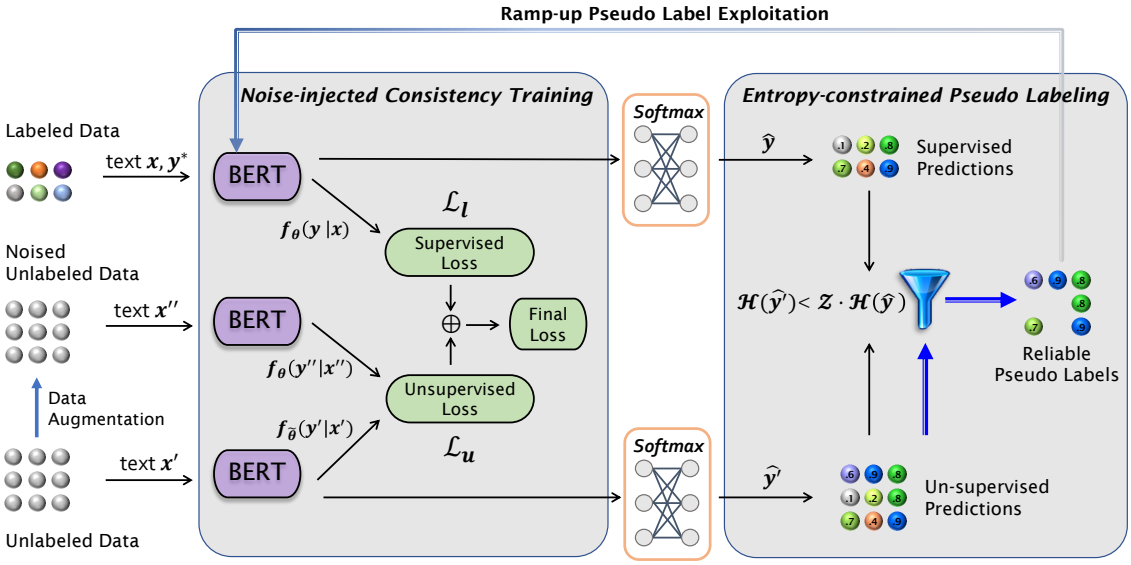


Figure 2: Illustration of **CPSUM** with Noise-injected Consistency Training and Entropy-constrained Pseudo Labeling for Exactive Summarization. In the figure, θ refers to the parameters of the model, and $\tilde{\theta}$ means it is just a copy of θ and the gradient will not propagate through it. $\hat{y}$ and $\hat{y'}$ refer to the logits by the softmax of labeled and unlabeled samples, respectively, $\mathcal{H}$ is the symbol for entropy, and $\mathcal{Z}$ is the normalization factor.

the entropy of the predicted result of unlabeled data is smaller than that of labeled data used at the current training step, then the low noisy pseudo labels generated by unlabeled data should be reserved.

Pseudo-label Selection. Either soft pseudo-labels selection (**Soft PlS**) or hard pseudo-labels selection (**Hard PlS**) can be adopted. We denote $logit$ and $logit'$ as the outputs of labeled and unlabeled data after the sigmoid operation by the model.

For soft pseudo labels, they are equivalent to the logit of unlabeled data, namely $y_{soft} = logit'$. Hard pseudo labels are essentially the binary vectors mapping soft pseudo labels to '0-1' space. Suppose that a document has K sentences which will be tagged with K labels. C represents the number of summary sentences. Then, hard pseudo labels can be defined as: $y_{hard} = f_{ope}(y_{soft}) \subseteq \{0,1\}^K$, where f_{ope} is the mapping operation converting the top-C probability of y_{soft} to '1' and the remaining to '0'.

Adaptive Entropy Constraint. Unlike text classification problems focusing on the highest probability (Kim, 2014), extractive summarizations must consider the multiple highest probabilities. This situation makes the selection criteria based on the confidence threshold hard to work.

Intuitively, pseudo labels with smaller entropy contribute more to the training process (Grandvalet and Bengio, 2004). We can leverage the entropy of labeled data to constrain that of unlabeled data, thus adapting the unlabeled data to model training. The noise introduced by low-confidence pseudo labels will be avoided through the entropy constraint, which further ensures the performance of the pseudo-labeling method. Besides, the adaptive constraint is more available due to eliminating manual settings.

To be specific, at training step t, assuming that labeled data x and unlabeled data x' are adopted, their respective outputs after the sigmoid operation by the model are $logit$ and $logit'$(namely y and y'), $\hat{y}$ and $\hat{y'}$ are labeled and unlabeled logits after the softmax operation. We denote K as the number of sentences in a document, then the labeled entropy $\mathcal{H}(\hat{y})$ and the unlabeled entropy $\mathcal{H}(\hat{y'})$ can be calculated as follows:

$$\begin{aligned} \mathcal{H}(\hat{y}) &= \sum_{k=1}^{K} \hat{y_k} \log(\hat{y_k}), \\ \mathcal{H}(\hat{y'}) &= \sum_{k=1}^{K} \hat{y'_k} \log(\hat{y'_k}), \end{aligned} \tag{3}$$

where k-th element of $\hat{y}$ and $\hat{y'}$ can be denoted as:

$$\begin{aligned} \hat{y_k} &= \frac{e^{logit_k}}{\sum_{k=1}^{K}(e^{logit_k})}, \\ \hat{y'_k} &= \frac{e^{logit'_k}}{\sum_{k=1}^{K}(e^{logit'_k})}. \end{aligned} \tag{4}$$

Then, the final selection constraint would be:

$$sel(y') = \begin{cases} 1, & \mathcal{H}(\hat{y'}) < \mathcal{Z} \cdot \mathcal{H}(\hat{y}), \\ 0, & otherwise, \end{cases} \quad (5)$$

where $sel(y') = 1$ means that y' is selected as a pseudo label and 0 otherwise. $\mathcal{Z}$ is the normalization factor and $\mathcal{Z} = dim_{x'}/dim_x$, aimed at alleviating the error propagation caused by inconsistent output vectors dimensions. $dim_{\hat{y'}}$ and $dim_{\hat{y}}$ denote the dimensions of $\hat{y'}$ and $\hat{y}$ respectively.

Ramp-up Pseudo-labels Exploitation. Although the reliability of entropy-constrainted pseudo-labels has been improved, the downside is that noises in them still exist, especially early in training. To mitigate this issue, we set a linearly increasing probability p_t in each epoch to select the filtered pseudo labels again:

$$p_t = min(1, \frac{t}{\tau_{epoch}}), \quad (6)$$

where t is the current epoch, and p_t denotes that the pseudo labels filtered in epoch t will be selected with probability p_t. τ_{epoch} is a hyper-parameter, which can be set according to the labeled data size.

For the way of adding pseudo labels to the labeled dataset, we draw on the idea of early stopping (Prechelt, 1996). As shown in the Algorithm 1 (lines 15-21), if ROUGE evaluated on the validation set for three consecutive rounds is in a downward trend, all filtered pseudo labels will be added to the labeled dataset. This procedure can effectively prevent the over-fitting phenomenon caused by repeated training of limited labeled data.

Algorithm 1 Training Procedure for Consistency Learning and Pseudo Labeling

Input: (x, y^*): the labeled data pair.
x' and x'': the unlabeled data and its augmented data.
R: the ratio of unlabeled and labeled data in each training step.
$step_{total}$: the total training step.
$val-interval$: the cycle of validation.

1: $t \leftarrow 0$; $psd \leftarrow []$; $val \leftarrow []$; $p_t \leftarrow 0$
2: **while** $t < step_{total}$ **do**
3: y is the output of x after $sigmoid$ by the model
4: $\hat{y} \leftarrow$ Eq. 4, $\mathcal{H}(\hat{y}) \leftarrow$ Eq. 3
5: **for** $r \leftarrow 1$ to R **do**
6: y' is the output of x' after $sigmoid$ by the model
7: $\hat{y'} \leftarrow$ Eq. 4, $\mathcal{H}(\hat{y'}) \leftarrow$ Eq. 3
8: $\mathcal{Z} \leftarrow dim_{\hat{y'}}/dim_{\hat{y}}$
9: **if** $\mathcal{H}(\hat{y'}) < \mathcal{Z} \cdot \mathcal{H}(\hat{y})$ **then**
10: Append (x', y') to psd according to p_t
11: **end if**
12: **end for**
13: $\mathcal{L}_l \leftarrow$ Eq. 1; $\mathcal{L}_u \leftarrow$ Eq. 2; $p_t \leftarrow$ Eq. 6
14: $\mathcal{L} \leftarrow \mathcal{L}_l + \omega(t)\mathcal{L}_u$, update the model
15: **if** t % $val-interval$ is 0 **then**
16: Append ROUGE of validation sets to val
17: **end if**
18: **if** the last three values of val monotonically decrease **then**
19: Merge pse to the labeled dataset
20: Stop pseudo-labels exploitation
21: **end if**
22: $t \leftarrow t + 1$
23: **end while**

4 Experimental Setup

4.1 Datasets

We conduct experiments on the following two datasets: (1) **CNN/DailyMail** (Hermann et al., 2015) includes news articles and corresponding extractive highlights. We use the standard splits (Hermann et al., 2015) for validation and testing. (2) **BBC XSum** (Narayan et al., 2018) provides a high level of abstraction. It has one-sentence summaries and is more abstractive than the CNN/DailyMail dataset. We obtain both labeled and unlabeled data from the entire dataset. Specifically, we divide parts of the original dataset into labeled data. For the rest, we delete the labels and treat them as unlabeled data.

4.2 Baselines and Evaluation Metrics

We focus on leveraging BERTSUMEXT (Liu and Lapata, 2019; Liu, 2019) for summarization [2]. To verify the effectiveness of our semi-supervised learning method in low-resource scenarios, we compare our method with the BERTSUMEXT of supervised learning. We also release the rule-based baselines — LEAD-3 on the CNN/DailyMail dataset and LEAD-1 on the BBC XSum dataset(excluding

[2]BERTSUMEXT is the variant of BERT, which builds several summarization-specific layers stacked on top of the BERT outputs including Simple Classifier, Transformer, and RNN. Our experiments mainly adopt BERT with the plainest Simple Classifier layer due to insignificant performance differences among the three layers.

Method	Labeled Data	*CNN/Daliy Mail*								
		ROUGE-1			ROUGE-2			ROUGE-L		
		P	R	F1	P	R	F1	P	R	F1
ORACLE		*43.63*	*58.77*	*48.35*	*23.88*	*31.77*	*26.28*	*40.30*	*54.13*	*44.61*
LEAD-3		34.50	51.94	40.04	14.81	22.44	17.21	31.16	46.86	36.14
Supervised (BERT)	10	31.47	41.92	34.42	11.84	15.65	12.89	28.29	37.61	30.91
	100	34.60	49.93	39.35	14.80	21.37	16.81	31.25	45.03	35.51
	1000	35.06	52.30	40.38	15.33	22.93	17.70	31.75	47.32	36.69
CPSUM w. Soft PlS	10	31.53	42.44	34.62	11.45	15.58	13.09	28.28	38.06	31.10
	100	34.77	52.04	40.22	15.02	22.56	17.37	31.42	46.96	36.32
	1000	35.17	53.23	40.93	15.48	23.49	18.01	31.85	48.14	37.04
CPSUM w. Hard PlS	10	31.69	42.78	34.94	11.63	15.78	13.26	28.58	38.33	31.37
	100	34.92	52.57	40.52	15.23	22.98	17.67	31.56	47.45	36.60
	1000	35.21	53.26	41.02	15.51	23.53	18.08	31.89	48.16	37.10
Method	**Labeled Data**	***BBC XSum***								
		ROUGE-1			ROUGE-2			ROUGE-L		
		P	R	F1	P	R	F1	P	R	F1
ORACLE		*31.37*	*30.51*	*29.57*	*9.74*	*9.07*	*8.86*	*22.96*	*22.07*	*21.47*
LEAD-1		17.12	16.69	16.32	1.68	1.66	1.60	12.59	12.24	11.96
Supervised (BERT)	10	18.21	15.92	16.13	2.17	1.86	1.90	13.68	11.81	12.01
	100	18.06	16.49	16.43	2.20	1.90	1.93	13.84	11.85	12.14
	1000	18.23	17.55	16.94	2.28	2.24	2.14	13.52	12.82	12.44
CPSUM w. Soft PlS	10	18.27	15.98	16.23	2.21	1.95	1.95	13.70	11.82	12.04
	100	18.75	16.54	16.82	2.29	2.02	2.04	13.96	12.10	12.35
	1000	18.95	17.35	17.22	2.38	2.22	2.17	14.05	12.77	12.71
CPSUM w. Hard PlS	10	18.29	15.98	16.25	2.21	1.96	1.95	13.68	11.83	12.05
	100	18.79	16.57	16.93	2.31	2.05	2.08	14.00	12.15	12.40
	1000	18.97	17.41	17.29	2.39	2.26	2.18	14.09	12.76	12.73

Table 1: Low-resource performance of ROUGE results on **CNN/DailyMail** and **BBC XSum** dataset. The best results for each group on all target corpora with 10, 100, and 1000 labeled examples are in-bold.

the one-line summary (Narayan et al., 2018)). The ground truth labels, which we call ORACLE, are extracted using the greedy approach. We use 10/100/1000 labeled data for supervised learning, and evaluate the summarization performance by ROUGE (Lin, 2004) in this paper, where R-1, R-2, and R-L are variants used to measure the overlap of unigrams, bigrams, and longest common subsequences between system and reference summaries.

4.3 Implementation Details

During training on the CNN/DailyMail dataset, the documents are truncated to 512 tokens, and the summaries are limited to 128 tokens. These two numbers are 512 and 64 for the BBC XSum dataset. Generally, semi-supervised learning performs a larger data size on unlabeled data than labeled data to fully use large quantities of unlabeled data. Therefore, we feed 1 batch of labeled data and 4 batches of unlabeled data into the framework in each training step, which is found to perform effectively by implementing the different proportions of labeled and unlabeled data. We use a batch size of 4 for labeled and unlabeled data.

We set τ_{epoch} to 30/15/5 epochs for training 10/100/1000 labeled data. All models are trained for 500/5000/20000 steps with 10/100/1000 labeled data on 3 Tesla V100 GPUs. The learning rate starts at 2e-3 and decay every 1000 steps. We also perform a linear warmup method to increase the learning rate smoothly from 0 to 2e-3 during 2000 steps at the beginning of training.

5 Experimental Results

5.1 Results on Dataset with long Summaries

The upper part of Table 1 shows the results on the CNN/DailyMail dataset. As shown, the model's performance improves as more labeled data becomes available. In the case where the data size is 1000, our method CPSUM achieves a +0.89 point improvement in R-1 and a +0.90 point improvement in R-L, compared with the traditional rule-based LEAD-3 on the CNN/DailyMail dataset.

Compared with the supervised baseline, CPSUM performs salient in all data sizes. The improve-

Table 2: **Ablation study on the effectiveness enhancement with ROUGE (F1) from different components based on CNN/DailyMail and BBC XSum dataset of 100 labeled data with hard pseudo labels**, including supervised learning, consistency regularization (CR), all pseudo labels (PlS [all]), and filtered pseudo labels (PlS [filtered]).

Components				Datasets					
				CNN/Daily Mail			*XSum*		
Supervised	CR	PlS [all]	PlS [filtered]	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
✓				39.35	16.81	35.51	16.43	1.93	12.14
✓	✓			39.62	16.94	35.79	16.52	1.96	12.04
✓	✓	✓		39.39	16.68	35.40	16.15	1.90	12.00
✓	✓		✓	**40.52**	**17.67**	**36.60**	**16.93**	**2.08**	**12.40**

Table 3: The number of pseudo labels w/o or w. ([all] or [filtered]) entropy-constrained filtering and the epochs to start appending pseudo labels for training.

CNN/Daily Mail					
Labeled Data	PlS [all]		PlS [filtered]		Epoch
10	960	9600%	248	2480%	24
100	9600	9600%	248	2480%	24
1000	28000	2800%	4675	467.5%	7
BBC XSum					
Labeled Data	PlS [all]		PlS [filtered]		Epoch
10	1200	12000%	288	2880%	30
100	12000	12000%	4043	4043%	30
1000	44000	4400%	11737	1173.7%	11

ments are most evident in the case of 100 labeled data with hard pseudo labels, where R-1/R-2/R-L increase +1.17/+0.86/+1.09 points compared with the supervised ones, far exceeding the baseline performance. These results all indicate the effectiveness of CPSUM in low-resource scenarios.

Moreover, soft or hard PlS also affects performance differently. We take 100 labeled data as an example. Although when soft labels are selected, CPSUM has obtained a +0.87 point improvement in R-1 compared with supervised learning, the adoption of hard pseudo labels still allows R-1 to continue to increase a +0.30 point upon soft pseudo labels. The results occur probably because soft labels are more ambiguous and have less information for extractive summarization than hard labels, so the use of hard pseudo labels will result in better performance than soft labels.

5.2 Results on Dataset with Short Summaries

We also conduct experiments to verify if CPSUM would be equally effective on the abstractive BBC XSum dataset, as shown in the lower part of Table 1. Identically, CPSUM outperforms supervised learning in all data sizes. For the better-performing hard pseudo labels, when labeled data sizes are 100 and 1000, CPSUM achieves remarkable performance. When there are only 10 labels, the performances of soft and hard PIS are indifferent, but they are better than the supervised method and LEAD-1. The results on XSum represent that CPSUM is also effective in generating extractive summaries.

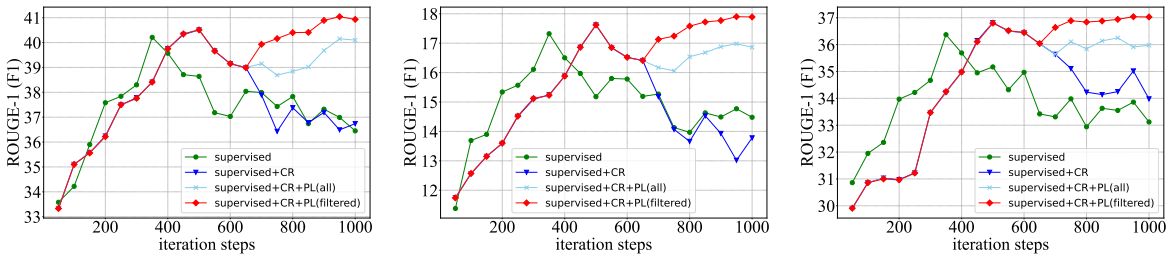
5.3 Analysis and Discussion

The Size of Selected Pseudo Labels. Table 3 shows the number of pseudo labels and the epochs to start adding pseudo labels with different labeled data sizes. In the case where labeled data sizes are 10 and 100, the numbers of filtered pseudo labels are much larger than that of labeled data.

Nevertheless, when the labeled data size is 1000, the filtered pseudo labels increase by only 467% relative to labeled data on the CNN/DailyMail dataset, which means the demand for pseudo data can be relatively reduced in the case of training on 1000 labeled data. The condition occurs because when the labeled data size is large, the speed of convergence is faster compared with a smaller labeled data size, according to the epoch determined by the pseudo label exploitation strategy (shown in the last column of Table 3).

Various Components Study. This study aims to verify the effectiveness enhancement of different components, including supervised learning, consistency regularization, all pseudo labels, and filtered pseudo labels in Table 2 . All the ablations are conducted with 100 labeled data in both datasets.

We take the CNN/DailyMail dataset as an example for analysis. The purely supervised learning is treated as the baseline, which achieves R-1/R-2/R-L of 39.35/16.81/35.51. After adding the consistency regularization method, R-1/R-2/R-L slightly have increased by +0.27/+0.13/+0.28 points. This means that although consistent learning will improve supervised learning, there are still factors that limit its performance, possibly the disadvanta-



(a) ROUGE-1 (b) ROUGE-2 (c) ROUGE-L

Figure 3: **The first 1000 steps change curve of ROUGE (F1) metrics on the CNN/DailyMail dataset of 100 labeled data.** All the experiments are validated every 50 steps. Before adding pseudo labels, the performance of consistency learning achieves the best at step 500(epoch 20), and subsequently, three consecutive validation results monotonically decrease, so we add pseudo labels at step 650. In all three metrics line charts, the red lines and skyblue lines rise after step 650 due to the addition of pseudo labels.

Table 4: **Ramp-up hyper-parameters τ_{epoch} exploration based on the CNN/DailyMail dataset of 100 labeled data with hard pseudo labels**.

τ_{epoch}	PlS size	ROUGE-1	ROUGE-2	ROUGE-L
10	4273	40.46	17.63	36.54
12	4072	40.47	17.64	36.55
15	3514	**40.52**	**17.67**	**36.60**
18	3698	40.47	17.64	36.55
20	3514	40.46	17.60	36.53
22	3316	40.41	17.55	36.46

geous noise of the augmented data.

Further, we compare two programs of selecting pseudo labels for labeled samples exploitation. First, all the pseudo labels are selected without filtering (in the 3-rd row), which obtains a slight improvement in R-1 but declines in R-2/R-L due to plenty of unreliable pseudo labels. The final method, which combines all our contributions, including consistency regularization and filtered pseudo labels (in the 4-th row), achieves superior results with +1.17/+0.86/+1.09 improvements in R-1/R-2/R-L, compared with the supervised baseline.

Fig. 3 shows the changing trend of ROUGE of all the ablations in the validation set during the first 1000 steps. As shown by the curve corresponding to the pseudo-labeling method, our frameworks with pseudo labels can sufficiently alleviate the overfitting caused by few-sample data. Additionally, Compared with not filtering reliable pseudo labels, the entropy-constrained method enables the model to improve upon the baseline effectively.

Ramp-up hyper-parameters Exploration. Our entropy-constrained pseudo labeling method introduces a hyper-parameter τ_{epoch} in the procedure of the ramp-up pseudo-labels exploitation. We tweak the hyper-parameter in a rational range and select it based on the CNN/DailyMail dataset with 100 labeled data. As shown in Fig. 4, results show that when τ_{epoch} is 15, CPSUM performs best. We find that in the supervised learning with 100 labeled data, CPSUM achieves the best in the 20-th epoch on the validation set. This indicates that in the case where the hyper-parameter τ_{epoch} is slightly smaller than the optimal training epoch, the performance will be the best. If τ_{epoch} is too tiny, the pseudo label data increase but become noisier. If τ_{epoch} is too large, the pseudo label data will decrease, and then high-quality pseudo labels will also decrease.

6 Conclusions

In this paper, we present a new perspective on effectively using consistency training and pseudo labeling to improve low resource extractive summarization over an insufficiently labeled dataset. With substituting simple noise injection operations with advanced data augmentation and constraining pseudo label selection with average entropy, our method brings substantial improvements compared with the supervised learning frameworks. Since our proposed model is orthogonal to the methods that using pre-trained models, we believe our model can be further boosted by taking other salient pre-trained models to initialize the text representations. Additionally, although we use ramp-up exploitation to control the adverse entropy effect brought by the early model, incorrect prediction cannot be avoided. An impeccable minimum entropy regularization method can be exploited in the future.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (No.U20B2053).

References

Philip Bachman, Ouais Alsharif, and Doina Precup. 2014. Learning with pseudo-ensembles. In *NeurIPS*, pages 3365–3373.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. In *NeurIPS*, pages 529–536.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NeurIPS*, pages 1693–1701.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling BERT for natural language understanding. In *EMNLP (Findings)*, volume EMNLP 2020, pages 4163–4174. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751. Association for Computational Linguistics.

Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. In *ICLR, poster*. OpenReview.net.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chen Liu, Mengchao Zhang, Zhibing Fu, Panpan Hou, and Yu Li. 2021a. Flitext: A faster and lighter semi-supervised text classification with convolution networks. In *EMNLP*, pages 2481–2491. Association for Computational Linguistics.

Yang Liu. 2019. Fine-tune BERT for extractive summarization. *CoRR*, abs/1903.10318.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *EMNLP-IJCNLP*, pages 3728–3738. Association for Computational Linguistics.

Ye Liu, Jianguo Zhang, Yao Wan, Congying Xia, Lifang He, and Philip S. Yu. 2021b. HETFORMER: heterogeneous transformer with sparse attention for long-text extractive summarization. In *EMNLP*, pages 146–154. Association for Computational Linguistics.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2019. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1979–1993.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, pages 3075–3081. AAAI Press.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *EMNLP*, pages 1797–1807. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543. Association for Computational Linguistics.

Lutz Prechelt. 1996. Early stopping-but when? In *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, pages 55–69. Springer.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh Singh Rawat, and Mubarak Shah. 2021. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *ICLR*. OpenReview.net.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *ICLR, Workshop*. OpenReview.net.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. 2019. Interpolation consistency training for semi-supervised learning. In *IJCAI*, pages 3635–3641. IJCAI.org.

Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. 2022. Semi-supervised semantic segmentation using unreliable pseudo-labels. *CoRR*, abs/2203.03884.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional BERT contextual augmentation. In *ICCS (4)*, volume 11539 of *Lecture Notes in Computer Science*, pages 84–95. Springer.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. 2020a. Unsupervised data augmentation for consistency training. In *NeurIPS*.

Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. 2020b. Self-training with noisy student improves imagenet classification. In *CVPR*.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *ACL*, pages 6197–6208. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019a. Searching for effective neural extractive summarization: What works and what's next. In *ACL , Volume 1: Long Papers*, pages 1049–1058. Association for Computational Linguistics.

Ming Zhong, Danqing Wang, Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2019b. A closer look at data bias in neural extractive summarization models. *CoRR*, abs/1909.13705.

Qingyu Zhou, Furu Wei, and Ming Zhou. 2020. At which level should we extract? an empirical analysis on extractive document summarization. In *COLING*, pages 5617–5628. International Committee on Computational Linguistics.

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. In *ACL (1)*, pages 654–663. Association for Computational Linguistics.