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

Yiming Wang\textsuperscript{1,2,3,†}, Qianren Mao\textsuperscript{1,2,†}, Junnan Liu\textsuperscript{1,2}, Weifeng Jiang\textsuperscript{1,2}, Hongdong Zhu\textsuperscript{1,2}, Jianxin Li\textsuperscript{1,2,*}

\textsuperscript{1} Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, China.
\textsuperscript{2} The State Key Laboratory of Software Development Environment, Beihang University, China.
\textsuperscript{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-
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).
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. U²PL (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 $f_\theta$ to represent the distributions of model predictions, where $\theta$ 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:

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

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 $f_\theta(y|x')$ 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:

$$L_u = \frac{1}{|X_u|} \sum_{(x', x'') \in X_u} l(f_\theta(y'|x'), f_\theta(y''|x'')), \quad (2)$$

where $X_u$ is a set of pairs containing $|X_u|$ unlabeled data $x'$ and the corresponding augmented data $x''$, and $\theta$ is just a copy of the current parameters $\theta$ 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: $L_f = L_l + w(t)L_u$, 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
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 PIS) or hard pseudo-labels selection (Hard PIS) can be adopted. We denote \( \text{logit} \) and \( \text{logit}' \) 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 \( y_{soft} = \text{logit}' \).

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: \( y_{hard} = f_{\text{ope}}(y_{soft}) \subseteq \{0, 1\}^K \), where \( f_{\text{ope}} \) 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 \( \hat{y} \) and \( \hat{y}' \) (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:

\[
\mathcal{H}(\hat{y}) = \sum_{k=1}^{K} y_k \log(\hat{y}_k),
\]

\[
\mathcal{H}(\hat{y}') = \sum_{k=1}^{K} y'_k \log(\hat{y}'_k),
\]

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

\[
\hat{y}_k = \frac{e^{\text{logit}_k}}{\sum_{k=1}^{K} e^{\text{logit}_k}},
\]

\[
\hat{y}'_k = \frac{e^{\text{logit}'_k}}{\sum_{k=1}^{K} e^{\text{logit}'_k}}.
\]

**Figure 2:** Illustration of CPSUM with Noise-injected Consistency Training and Entropy-constrained Pseudo Labeling for Extractive Summarization. In the figure, \( \theta \) refers to the parameters of the model, and \( \tilde{\theta} \) means it is just a copy of \( \theta \) 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 \( Z \) is the normalization factor.
Then, the final selection constraint would be:

\[
\text{sel}(y') = \begin{cases} 
1, & \mathcal{H}(y') < Z \cdot \mathcal{H}(\hat{y}), \\
0, & \text{otherwise},
\end{cases}
\]  

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

**Ramp-up Pseudo-labels Exploitation**. Although the reliability of entropy-constrained 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:

\[
p_t = \min(1, \frac{t}{\tau_{\text{epoch}}}),
\]

where \( t \) is the current epoch, and \( p_t \) denotes that the pseudo labels filtered in epoch \( t \) will be selected with probability \( p_t \). \( \tau_{\text{epoch}} \) 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.

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.

**Algorithm 1** Training Procedure for Consistency Learning and Pseudo Labeling

**Input**: \((x, y^s)\): the labeled data pair. \( x' \) and \( y'^s \): the unlabeled data and its augmented data. 

- \( R \): the ratio of unlabeled and labeled data in each training step.
- \( \text{step}_{\text{total}} \): the total training step.
- \( \text{val-interval} \): the cycle of validation.

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

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.
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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Labeled Data</th>
<th>CNN/DailyMail</th>
<th></th>
<th></th>
<th></th>
<th>BBC XSum</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ROUGE-1</td>
<td>ROUGE-2</td>
<td>ROUGE-L</td>
<td>ROUGE-1</td>
<td>ROUGE-2</td>
<td>ROUGE-L</td>
<td>ROUGE-1</td>
<td>ROUGE-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P</td>
<td>R</td>
<td>F1</td>
<td>P</td>
<td>R</td>
<td>F1</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>ORACLE</td>
<td></td>
<td>43.63</td>
<td>58.77</td>
<td>48.35</td>
<td>23.88</td>
<td>31.77</td>
<td>26.28</td>
<td>40.30</td>
<td>54.13</td>
</tr>
<tr>
<td>LEAD-3</td>
<td></td>
<td>34.50</td>
<td>59.14</td>
<td>40.04</td>
<td>14.81</td>
<td>22.44</td>
<td>17.21</td>
<td>31.16</td>
<td>46.86</td>
</tr>
<tr>
<td>Supervised (BERT)</td>
<td>10</td>
<td>31.47</td>
<td>41.92</td>
<td>34.42</td>
<td>11.84</td>
<td>15.65</td>
<td>12.89</td>
<td>28.29</td>
<td>37.61</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>34.60</td>
<td>49.93</td>
<td>39.35</td>
<td>14.80</td>
<td>21.37</td>
<td>16.81</td>
<td>31.25</td>
<td>45.03</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>35.06</td>
<td>52.30</td>
<td>40.38</td>
<td>15.33</td>
<td>22.93</td>
<td>17.70</td>
<td>31.75</td>
<td>47.32</td>
</tr>
<tr>
<td>CPSUM w. Soft PIS</td>
<td>10</td>
<td>31.53</td>
<td>42.44</td>
<td>34.62</td>
<td>11.45</td>
<td>15.58</td>
<td>13.09</td>
<td>28.28</td>
<td>38.06</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>34.77</td>
<td>52.04</td>
<td>40.22</td>
<td>15.02</td>
<td>22.56</td>
<td>17.37</td>
<td>31.42</td>
<td>46.96</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>35.17</td>
<td>53.23</td>
<td>40.93</td>
<td>15.48</td>
<td>23.49</td>
<td>18.01</td>
<td>31.85</td>
<td>48.14</td>
</tr>
<tr>
<td>CPSUM w. Hard PIS</td>
<td>10</td>
<td>31.69</td>
<td>42.78</td>
<td>34.94</td>
<td>11.63</td>
<td>15.78</td>
<td>13.26</td>
<td>28.58</td>
<td>38.33</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>34.92</td>
<td>52.57</td>
<td>40.52</td>
<td>15.23</td>
<td>22.98</td>
<td>17.67</td>
<td>31.56</td>
<td>47.45</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>35.21</td>
<td>53.26</td>
<td>41.02</td>
<td>15.51</td>
<td>23.53</td>
<td>18.08</td>
<td>31.89</td>
<td>48.16</td>
</tr>
</tbody>
</table>

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 \( \tau_{epoch} \) 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]).

<table>
<thead>
<tr>
<th>Components</th>
<th>Datasets</th>
<th>CNN/Daily Mail</th>
<th>XSum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ROUGE-1</td>
<td>ROUGE-2</td>
</tr>
<tr>
<td>Supervised</td>
<td></td>
<td>39.35</td>
<td>16.81</td>
</tr>
<tr>
<td>CR</td>
<td></td>
<td>39.62</td>
<td>16.94</td>
</tr>
<tr>
<td>PlS [all]</td>
<td></td>
<td>39.39</td>
<td>16.68</td>
</tr>
<tr>
<td>PlS [filtered]</td>
<td></td>
<td>40.52</td>
<td>17.67</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Labeled Data</th>
<th>CNN/Daily Mail</th>
<th>BBC XSum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PlS [all]</td>
<td>PlS [filtered]</td>
</tr>
<tr>
<td>10</td>
<td>960 9600%</td>
<td>248 2480%</td>
</tr>
<tr>
<td>100</td>
<td>9600 9600%</td>
<td>248 2480%</td>
</tr>
<tr>
<td>1000</td>
<td>28000 28000%</td>
<td>4675 467.5%</td>
</tr>
<tr>
<td></td>
<td>1200 1200%</td>
<td>288 2880%</td>
</tr>
<tr>
<td>100</td>
<td>12000 12000%</td>
<td>4043 4043%</td>
</tr>
<tr>
<td>1000</td>
<td>44000 4400%</td>
<td>11737 1173.7%</td>
</tr>
</tbody>
</table>

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 PlS 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.

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.

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-
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 $\tau_{epoch}$ exploration based on the CNN/DailyMail dataset of 100 labeled data with hard pseudo labels.

<table>
<thead>
<tr>
<th>$\tau_{epoch}$</th>
<th>PIS size</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4273</td>
<td>40.46</td>
<td>17.63</td>
<td>36.54</td>
</tr>
<tr>
<td>12</td>
<td>4072</td>
<td>40.47</td>
<td>17.64</td>
<td>36.55</td>
</tr>
<tr>
<td>15</td>
<td>3514</td>
<td><strong>40.52</strong></td>
<td><strong>17.67</strong></td>
<td><strong>36.60</strong></td>
</tr>
<tr>
<td>18</td>
<td>3698</td>
<td>40.47</td>
<td>17.64</td>
<td>36.55</td>
</tr>
<tr>
<td>20</td>
<td>3514</td>
<td>40.46</td>
<td>17.60</td>
<td>36.53</td>
</tr>
<tr>
<td>22</td>
<td>3316</td>
<td>40.41</td>
<td>17.55</td>
<td>36.46</td>
</tr>
</tbody>
</table>

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 $\tau_{epoch}$ 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 $\tau_{epoch}$ 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 $\tau_{epoch}$ is slightly smaller than the optimal training epoch, the performance will be the best. If $\tau_{epoch}$ is too tiny, the pseudo label data increase but become noisier. If $\tau_{epoch}$ 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


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.


