# Prompt-based Pseudo-labeling Strategy for Semi-Supervised Extractive Summarization

**Anonymous ACL submission**


## Abstract

This work tackles the task of extractive text summarization in a limited labeled data scenario using a semi-supervised approach. Prior works have focused mainly on using the model's output probabilities to evaluate the pseudo-labels; however, these probabilities might not capture the semantics and correctness of the pseudo-labels in the way current LLMs (such as GPT-4) can because of their strong domain knowledge. Therefore, we propose a prompt-based pseudo-label selection strategy using an off-the-shelf LLM and show that a good selection strategy can result in competitive performance as the fully supervised setting. We evaluate our method on three text summarization datasets: TweetSumm, WikiHow, and ArXiv/PubMed. Our experiments show that by using an LLM to evaluate and generate pseudo-labels, we can improve the ROUGE-1 by 10-20% on the different datasets, without introducing any task-specific objective functions during the training of the summarization model. We also show that such a method needs a smaller pool of unlabeled examples to perform better.[1]


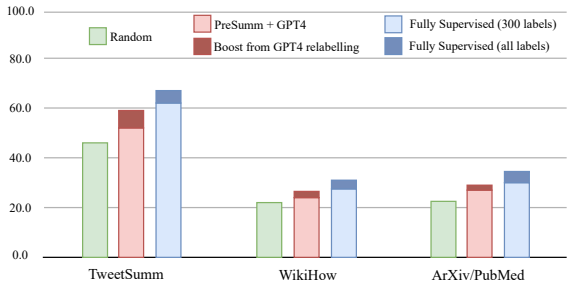


Figure 1: **Comparison of different pseudo-label selection strategies.** The demonstration shows Rouge-1 scores for different pseudo-label-selection strategies on three datasets. The proposed method (middle bar for each dataset) combines PreSumm's confidence with GPT4's knowledge to generate high-quality summaries for unlabeled documents. We show that it intuitively outperforms the random baseline and achieves the closest performance to a fully supervised extractive text summarization model (right bar for each dataset).

## 1 Introduction

Text summarization is a crucial problem in today's data-driven era. For instance, more and more enterprises now have customer service chatlogs and wish to create an automated text summarization system to enhance the user experience (Feigenblat et al., 2021). The internet, too, is flooded with an overwhelming number of articles to read. With the exponential growth of digital content, the need for automated summarization techniques that can swiftly process and distill information becomes paramount (Koupaee and Wang, 2018; Cohan et al., 2018). However, in real-life scenarios, we often do not have enough labeled data to employ a fully supervised summarization model. Therefore, we explore the landscape of semi-supervised methods to solve the task. While supervised methods heavily rely on annotated data, incorporating semi-supervised learning mitigates the burdens of obtaining large-scale labeled datasets, thereby extending the applicability of text summarization models to various domains and languages. More specifically, we tackle the task of extractive summarization, where the summary is constructed by copying lines of the input document (Radev et al., 2004; Li et al., 2006).

Semi-supervised approaches for text summarization are heavily under-explored. Earliest works on extractive summarization through semi-supervision include (Amini and Gallinari, 2001; Wong et al., 2008), where both works reformulate the task of extractive text summarization as text classification. First, they train an unsupervised language model to learn sentence representations. Next, they train that model on a limited labeled dataset on the binary classification task of predicting whether or not a

[1]We will release code for reproducibility upon acceptance.

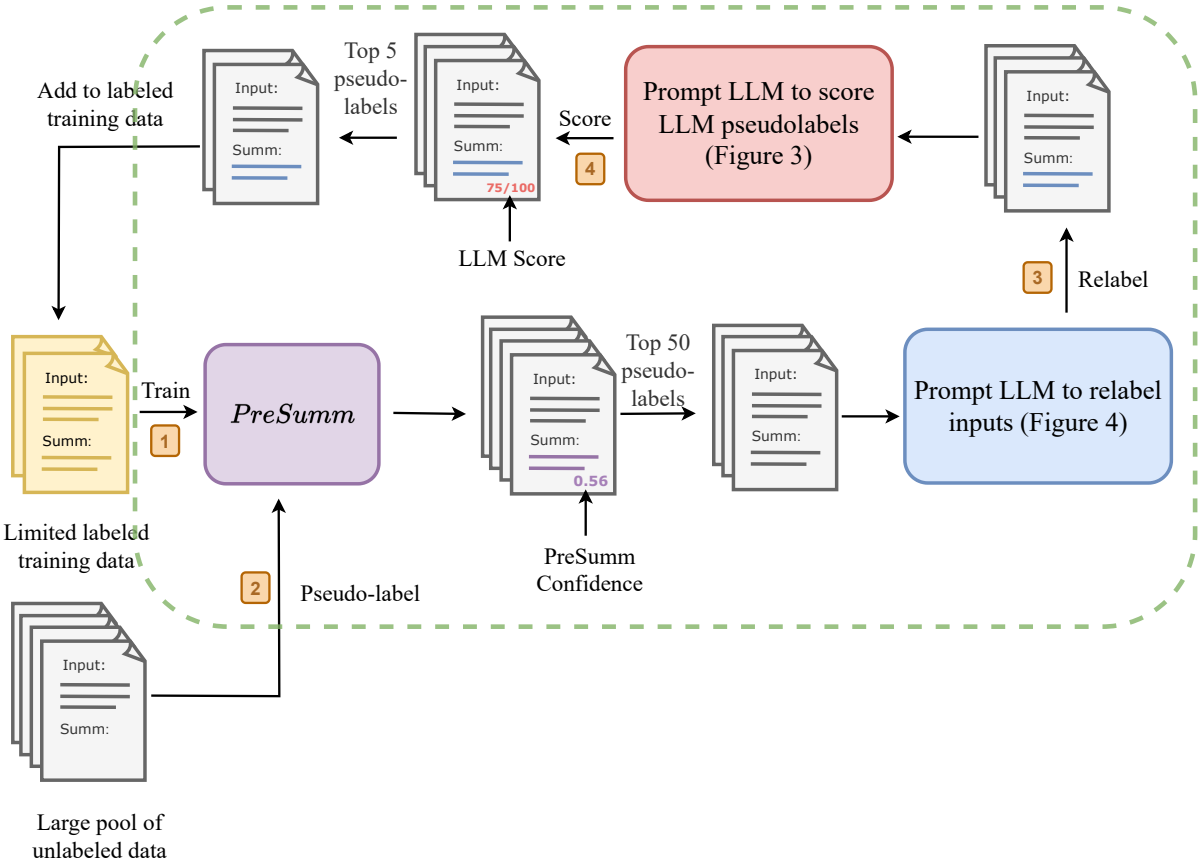


Figure 2: **Proposed method.** The dashed box highlights the pseudo-labeling and scoring method that combines PreSumm and an LLM. **Step 1:** we first train a PreSumm model on the limited labeled dataset **Step 2:** we then generate pseudo-labels using the same trained PreSumm model and select top 50 based on model confidence (refer to Equation 1). **Step 3:** Next, we prompt an off-the-shelf LLM like GPT-4 to construct an extractive summary of the top 50 documents from Step 2. **Step 4:** Finally, we generate a rating for the summaries in Step 3 by prompting an LLM and select the top 5 summaries. These summaries are then added to the training data for the next iteration.

sentence will be present in the summary. More recently, Liu and Lapata (2019) and Zhuang et al. (2023) employed a similar strategy using large language models (LLMs) like BERT as their backbones. While these works have shown promising results, they rely on model's probabilities and might not capture the semantics and correctness of the pseudo-labels in the way current LLMs (such as GPT-4) do due to their inherently strong domain knowledge.

Inspired by the recent success of using LLMs for generating and improving pseudo-labels for various tasks (Sahu et al., 2022, 2023; Dziri et al., 2023; Mishra et al., 2023), in this work, we focus on improving the process of selecting good-quality pseudo-labels during the training of the summarization model. Therefore, in this work, we propose a prompting-based pseudo-labeling strategy to use the unlabeled samples during the teacher-student learning phase. We first use the model's confidence to shortlist pseudolabels to show to the LLM, then we prompt the LLM to generate summaries for the shortlisted examples and then select the best summaries. We use GPT-4 (OpenAI, 2023) for our experiments, but our framework is compatible with any generative LLM. We evaluate our approach on three text summarization datasets: TweetSumm (Feigenblat et al., 2021), WikiHow (Koupaee and Wang, 2018), and ArXiv/PubMed (Cohan et al., 2018). We show that our method can enhance the performance of a standard teacher-student learning algorithm, which primarily relies on the teacher's confidence to determine a pseudo-label's quality. To the best of our knowledge, we are the first to show that a good selection and pseudo-labeling strategy can achieve improvements akin to developing a better pretrained model.

Our contributions can be summarized as follows:

1. We propose a prompt-based framework for

```
Input Prompt:
Below is an instruction that describes a task.
Write a response that appropriately completes
the request.

Text to summarize:
<Input Conversation/Text to Summarize>

Summary:
<Summary to evaluate>


Provide a rating between 0-100 based on the
following criteria:

1)The summary should be concise...
2)The summary should capture ...
3)The summary should be extractive, i.e.,..
4)The response should be strictly numerical.

Check very carefully before giving your rating.

Response:

85
```

Part 1

Part 2

Figure 3: **Prompt for scoring pseudo-labels.** The demonstration highlights the two main parts of the prompt: Part 1 contains the text and summary pair, and Part 2 lists the evaluation criteria. **Note:** for brevity, we do not include the full text from the evaluation criteria. Please refer to Section 2 for details about the evaluation criteria.

```
Input Prompt:
Below is an instruction that describes a task.
Write a response that appropriately completes
the request.

Text to summarize:
1. <Line 1 of the input text>
2. <Line 2 of the input text>
3. <Line 3 of the input text>
.
.
N. <Line N of the input text>

For each sentence in the input text, output
the probability of it appearing in the
summary such that the summary captures the key
points of the text. Output in the format
<line id>. <probability>.

Response:

1. 0.9
2. 0.1
3. 0.8
.
.
N. 0.2
```

Figure 4: **Prompt for generating pseudo-labels.** We attach a line ID for each sentence of the input text and instruct GPT-4 to use those line IDs in its response.

semi-supervised extractive text summarization.

2. Our generic framework uses an LLM (such as GPT4) to generate high-quality summaries for unlabeled examples.

3. We show that our method achieves state-of-the-art results on three datasets even when used in more challenging setups that prior works.

## 2 Methodology

This section describes our confidence-guided pseudo-label selection with GPT-4 score approach for training self-supervised summarization models. We divide the whole process into multiple stages, which are explained in detailed in the subsequent subsections. Figure 2 shows the entire pipeline.

### 2.1 Stage 1: Training the Teacher

We employ a standard teacher-student training framework during training. We train a fully-supervised model $M$ (teacher) on the limited labeled examples $D_l$ in the dataset and then use it to generate pseudo-labels for the unlabeled dataset $D_u$. We use PreSumm (Liu and Lapata, 2019) as our summarization model as it has been shown to perform well in the extractive setting. Therefore, we reformulate the task of generative extractive summaries to that of classification, where the model predicts whether or not the given sentence in the input text will be present in the output summary. We then construct the extractive summary from the top-$n$ sentences according to the model's predicted probability.

### 2.2 Stage 2: Generating Pseudo-labels using the Teacher

After training the teacher model, we pseudo-label the unlabeled examples $D_l$ and select a subset of 50 pseudo-labels with the highest teacher confidence. To define model confidence for a generated summary (pseudo-label), let $p_{ij}$ denote the probability of sentence $i$ in unlabeled datapoint $u_j$ being selected in the summary. We then compute model confidence ($C_j$) on an input text $u_j$ as follows:

$$C_j = \frac{\sum_{i=1}^{n}(p_{ij})}{n}, \quad (1)$$

where $n$ is the number of sentences in the generated summary (pseudo-label). We first shortlist a subset of 50 examples to avoid relabeling the large unlabeled pool and minimize our LLM usage cost in the next stages.

### 2.3 Stage 3: Relabeling using the LLM

After selecting the top 50 pseudo-labels using model confidence, we prompt GPT-4 to pseudo-label each unlabeled example. Figure 4 shows our prompt template for generating summaries using GPT-4. Note that we use the same mechanism when prompting as PreSumm since we want to generate extractive summaries.

### 2.4 Stage 4: Scoring using an LLM

Finally, we prompt the LLM to output a rating between 0-100 for all the pseudolabels from Stage 3, based on the following criteria:

- The summary should be concise, ideally with less than $n$ sentences.
- The summary should capture the key points from the input text. Do not penalize if it doesn't include any minor additional details.
- The summary should be extractive, i.e., directly taken from the conversation without any modifications or removal of symbols.
- The response should contain only the numerical rating.

Figure 3 shows our prompt template for scoring a pseudo-label. We then choose the top 5 unlabeled examples with the highest GPT-4 score for the pseudo-labels in Stage 2 and append them to the labeled training data for the next iteration.

### 2.5 Inference

We iterate Stages 1-4 multiple times to obtain a large pool of pseudo-labeled samples. Then, we use the summarization model obtained after the last iteration to summarize unseen documents from the test set. Algorithm 1 also summarizes the proposed semi-supervised approach for extractive text summarization.

**Algorithm 1:** An algorithm for prompt-based semi-supervised text summarization

**Input** : $D_i$, $D_u$, $N_{iter}$
**Output** : Trained summarization model $M$

$M \leftarrow null$ ;
**for** $i \leftarrow 1$ **to** $N_{iter}$ **do**
  `// Train PreSumm on labeled set;`
  $M \leftarrow PreSumm(D_l)$;
  `// Generate Pseudo-labels;`
  $Pseudos \leftarrow Top50(M(D_u))$;
  $Pseudos \leftarrow Top5(LLM(Pseudos))$;
  `// Add top-5 pseudos to labeled set and update unlabeled set;`
  $D_l \leftarrow D_l \cup Pseudos$;
  $D_u \leftarrow D_u \setminus Pseudos$;
**end**

## 3 Experimental Setup

This section describes our experimental setup, including the datasets used, training and evaluation details, and the baselines run.

### 3.1 Datasets

We use three datasets for extractive summarization. **1) TweetSumm** (Feigenblat et al., 2021) is a real-world customer service chat dataset. The dataset has 1100 conversations between a customer and an agent, and each conversation has three human-annotated extractive summaries. The training set has 858 dialogs, out of which we randomly sample a subset of 50 examples as the initial training set and treat the remaining as the unlabeled set. We use the official validation and test sets for evaluation.

**2) WikiHow** (Koupaee and Wang, 2018) contains WikiHow articles with their headlines as abstrative summaries. The dataset has over 180k articles, with around 168k training articles, and 6000 test and validation articles.

**3) ArXiv/PubMed** (Cohan et al., 2018) is a collection of scientific articles from PubMed and ArXiv with their abstracts as their summaries. The dataset has close to 325k articles, with nearly 300k training articles and 12.5k test and validation articles.

Notably, the WikiHow and ArXiv/PubMed datasets do not have extractive labels. Therefore, we follow the same preprocessing steps as the original PreSumm paper (Liu and Lapata, 2019) to convert the abstraction datasets to extractive datasets. Since the training corpus is humongous for WikiHow and ArXiv/PubMed, we limit the pool of unlabeled examples by randomly sampling a subset of 1,000 articles in each training cycle.

We choose the above three datasets to cover a diverse set of scenarios: TweetSumm contains real-world conversations and human-annotated extractive summaries, WikiHow contains medium-sized documents, and ArXiv/PubMed contains long documents and summaries.

| Method | TweetSumm | | | WikiHow | | | ArXiv/Pubmed | | |
|---|---|---|---|---|---|---|---|---|---|
| | R-1 (%) | R-2 (%) | R-L (%) | R-1 (%) | R-2 (%) | R-L (%) | R-1 (%) | R-2 (%) | R-L (%) |
| **DistilBERT$_{base}$** | | | | | | | | | |
| **Fully Supervised Models** | | | | | | | | | |
| PreSumm (50 labels) | 41.7 (1.6) | 32.4 (1.2) | 43.6 (2.1) | 19.2 (1.8) | 2.1 (0.6) | 9.1 (1.4) | 21.4 (1.2) | 4.7 (0.3) | 10.4 (1.2) |
| PreSumm (300 labels) | 58.5 (1.1) | 48.1 (1.7) | 57.8 (1.3) | 23.4 (1.1) | 6.3 (0.7) | 14.5 (1.4) | 25.9 (0.4) | 8.7 (0.2) | 16.4 (0.9) |
| PreSumm (all labels) | 65.8 (0.3) | 55.3 (0.7) | 63.3 (0.6) | 30.7 (0.2) | 8.6 (0.8) | 19.1 (0.1) | 34.2 (0.5) | 12.3 (0.6) | 19.2 (0.4) |
| **Semi-Supervised Models (50:250)** | | | | | | | | | |
| Random | 36.7 (1.5) | 25.4 (1.4) | 36.7 (1.3) | 19.7 (1.4) | 1.5 (1.1) | 7.2 (1.3) | 19.5 (1.3) | 2.9 (0.9) | 7.8 (1.2) |
| Confidence | 43.5 (1.4) | 35.1 (1.2) | 46.8 (1.1) | 21.3 (0.4) | 3.7(0.8) | 10.3 (1.0) | 23.4 (1.1) | 5.2 (0.7) | 12.5 (1.1) |
| + GPT4 relabeling | 55.4 (1.3) | 46.7 (0.6) | 56.1 (0.9) | 22.1 (0.4) | 5.7 (0.6) | 13.5 (0.7) | 23.8 (0.8) | 7.3 (1.3) | 15.3 (0.8) |
| Confidence w/ GPT4 score | 46.8 (1.3) | 37.4 (0.4) | 48.3 (1.2) | 21.7 (0.9) | 4.6 (0.4) | 12.1 (1.1) | 24.1 (0.9) | 6.7 (0.3) | 13.8 (1.4) |
| + GPT4 relabeling (Ours) | **57.6 (1.2)** | **46.3 (1.7)** | **56.2 (1.3)** | **22.7 (0.3)** | **5.9 (0.4)** | **13.8 (0.5)** | **24.7 (0.9)** | **8.1 (1.3)** | **15.9 (0.8)** |
| **BERT$_{base}$** | | | | | | | | | |
| **Fully Supervised Models** | | | | | | | | | |
| PreSumm (50 labels) | 48.7 (1.5) | 37.1 (1.1) | 47.9 (2.4) | 23.4 (1.9) | 3.2 (0.8) | 12.1 (1.1) | 26.4 (1.3) | 7.3 (0.9) | 13.5 (1.1) |
| PreSumm (300 labels) | 61.2 (1.5) | 51.1 (2.1) | 60.3 (1.7) | 26.6 (1.2) | 7.6 (0.6) | 16.7 (1.3) | 29.7 (0.8) | 10.8 (0.9) | 18.5 (1.2) |
| PreSumm (all labels) | 67.0 (0.7) | 58.1 (0.4) | 65.6 (0.5) | 31.1 (0.4) | 8.9 (0.9) | 19.1 (0.3) | 34.8 (0.6) | 12.3 (0.4) | 19.5 (0.5) |
| SSPT (50 labels) | 59.6 | 50.2 | 58.9 | - | - | - | - | - | - |
| **Semi-Supervised Models (50:250)** | | | | | | | | | |
| TSL (500:500) | 58.96 | 48.32 | 58.19 | - | - | - | - | - | - |
| TSL (50:500) | 49.0 | 37.7 | 48.2 | - | - | - | - | - | - |
| Random | 45.4 (1.4) | 32.4 (1.9) | 42.5 (1.8) | 22.1 (1.7) | 2.4 (1.5) | 9.6 (1.5) | 23.3 (1.4) | 6.1 (1.2) | 12.4 (1.3) |
| Confidence | 49.7 (1.6) | 39.5 (1.4) | 49.4 (1.3) | 24.5 (0.6) | 4.8 (1.1) | 12.8 (1.0) | 27.6 (1.1) | 7.7 (0.7) | 14.2 (1.2) |
| + GPT4 relabeling | 57.8 (1.2) | 50.3 (0.5) | 58.9 (1.2) | **26.4 (0.3)** | **7.3 (0.5)** | **16.4 (0.8)** | 28.7 (0.9) | 9.5 (1.1) | 17.1 (0.8) |
| Confidence w/ GPT4 score | 52.3 (1.6) | 42.8 (0.7) | 51.0 (1.4) | 25.2 (0.7) | 5.6 (0.5) | 13.1 (1.0) | 27.7 (0.9) | 7.9 (0.2) | 15.5 (1.3) |
| + GPT4 relabeling (Ours) | **58.9 (1.4)** | **50.4 (0.8)** | **59.4 (1.5)** | 26.1 (0.4) | 7.2 (0.6) | 15.9 (0.9) | **29.1 (0.7)** | **9.7 (1.2)** | **17.7 (0.6)** |

Table 1: Mean ROUGE F-1 Scores (with standard deviation) on the three datasets for three random seeds. R-1, R-2, and R-L denote ROUGE-1, ROUGE-2, and ROUGE-L metrics, respectively. The first block for DistilBERT$_{base}$ and BERT$_{base}$ shows the performance of fully-supervised models, while the second block compares different pseudo-label selection strategies. All semi-supervised models use 50 labeled and 250 unlabeled examples unless otherwise stated. SSPT (Mixed strategy), TSL (50:500), and TSL(500:500) results taken from Zhuang et al. (2023). **Note:** TSL (500:500) in BERT$_{base}$ is displayed separately because it uses 500 labeled and 500 unlabeled examples.

### 3.2 Training and Evaluation Details.

We use the TransformerSum repository[2] to implement our training pipeline using `distilbert-base-uncased` and `bert-base-uncased` as backbones for PreSumm architecture. We select 50 labeled examples as the initial training set and add 5 new pseudo-labels at the end of each cycle. We terminate the experiment after 50 cycles, giving a final dataset size of 300 (50 labeled and 250 pseudo-labeled examples). We set the summary size ($n$) to 4 for TweetSumm and 8 for WikiHow and ArXiv/PubMed We base these summary sizes on the average summary size of the labeled training set. We train all our models of 1 12GB V100 GPU.

We report the test ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L) F-1 scores (Lin, 2004) for the three datasets. ROUGE-1 and ROUGE-2 scores measure the $n$-gram overlap between the predicted and the ground truth summaries, whereas ROUGE-L also takes into consideration the order of $n$-grams. We run each experiment for three random seeds and report the mean and standard deviation in Table 1.

### 3.3 Baselines

We compare our method with the following baselines:

**1) PreSumm** (Liu and Lapata, 2019). The original PreSumm model that pretrains a BERT model for summarization. The BERT encoder creates sentence representations, which are then used to predict whether or not a sentence will be present in the summary. We train three PreSumm models – one on a limited training set with 50 labeled examples, another with 300 labeled examples, and one on the complete training set.

**2) Self-supervised Pre-training (SSPT)** (Zhuang

[2] https://transformersum.readthedocs.io/en/latest/

et al., 2023). Similar to PreSumm, where, instead of using vanilla BERT as the sentence encoder, it uses an enhanced BERT model obtained by pertaining on self-supervised tasks designed specifically for dialog summarization. Since the work focuses solely on dialogs, it only evaluates on the TweetSumm dataset. We report results for the "Mixed strategy" as it is shown to be the best model in the paper.

**3) Teacher-Student Learning (TSL)** (Zhuang et al., 2023). The teacher-student learning framework uses a slightly different formulation for model confidence than ours, as given below:

$$C_j = \sum_{i=1}^{n} (C_{ij})/n_j \tag{2}$$

In Equation 2, $C_{ij} = p_{ij}q_{ij} + (1 - p_{ij})(1 - q_{ij})$, where $p_{ij}$ is the probability of sentence $i$ being selected for summary for dialog $j$ estimated by the teacher model, and $q_{ij} = 1$ if $p_{ij}$ in top 4, else 0. Notably, this formulation also considers the teacher's confidence in the sentences it chooses to discard from the summary. We report the performance of TSL (50:500) as it was the closest to our setup (50 labeled examples and 500 unlabeled examples).

**4) Confidence-guided pseudo-label selection with GPT4 (Ours).** Our proposed method for selecting pseudo-labels using PreSumm and GPT-4 confidence, followed by pseudo-labeling with GPT-4. We denote this model by "Confidence w/ GPT4 score + GPT4 relabeling" in Table 1.

**5) Random.** We randomly add five pseudo-labels from the unlabeled set in each cycle instead of using model confidence and GPT4 relabeling as in **4)**.

# 4 Results

We now discuss results in Table 1, which contains two sets of results for DistilBERT$_{base}$ and BERT$_{base}$ as backbones for the PreSumm architecture. The first block for each backbone type shows the performance of fully supervised PreSumm models trained on three scenarios. The second block compares different strategies for selecting pseudo-labels in a 50:230 labeled:unlabeled semi-supervised setting.

## 4.1 Performance of Fully-supervised Models

Intuitively, we note that including more labels leads to better rouge scores on all the datasets; however, improvements on WikiHow and ArXiv/PubMed are less pronounced than on the TweetSumm dataset. This might be because the extractive labels for WikiHow and Arxiv/PubMed were created using a greedy heuristic that matches sentences from an abstractive summary with the input article. We must also note that the PreSumm (300 labels) model is a proxy for an oracle model, as it simulates the best-case scenario where all the pseudo-labels are high-quality. Finally, we observe that SSPT (50 labels) is comparable to PreSumm (300 labels), suggesting an efficient knowledge transfer for the enhanced encoder in SSPT. Additionally, we note that PreSumm with DistilBERT matches the performance of PreSummm with BERT when it has access to all labels (see Presumm (all labels) row).

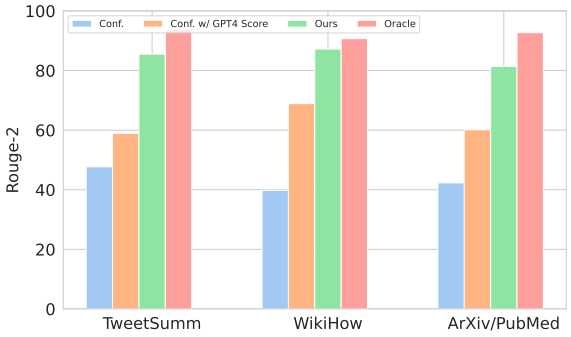


Figure 5: **Quality of selected pseudo-labels.** This figure compares the quality of pseudo-labels selected using different strategies. The y-axis denotes the ROUGE-2 scores of the top-5 pseudolabels for each strategy. **Note:** Results are for BERT$_{base}$ backbone. We obtain the oracle values by selecting pseudo-labels with the highest ROUGE-2 scores against the ground truth summaries.

## 4.2 Comparison of Selection Strategies

First, we note that all pseudo-label selection strategies for semi-supervised models outperform the random baseline. This affirms the importance of a good pseudo-label selection strategy, especially when we do not have access to a large pool of labeled examples.

The random baseline performs worse than the PreSumm (50 labels), suggesting that the majority of the pseudo-labels *selected* by the model in each are low-quality. Using model confidence leads to slight performance gains on all the datasets, and adding GPT4 score further improves the results (Confidence v/s Confidence w/ GPT4 score). This suggests that the summarization model also generates some high-quality pseudo-labels, and using GPT4 to rate those pseudo-labels is beneficial in

selecting such examples. Our small-scale study further confirms this, where we compute the ROUGE scores of the selected pseudo-labels against the ground truth summaries of the unlabeled examples. Figure 5 shows the results of the study. A stark difference between “Conf" and “Oracle" shows that relying solely on the model’s confidence consistently leads to low-quality pseudo-labels. We note that combining GPT4 score with model’s confidence helps improve the quality of selected pseudo-labels.

### 4.3 Effect of Relabeling

Relabeling with GPT-4 leads to significant boosts in ROUGE scores across the board (Confidence v/s Confidence + GPT4 relabeling and Confidence w/ GPT4 score + GPT relabeling). When using BERT$_{base}$ as the backbone, we note that the ROUGE-1 improves from 52.3 to 58.7 on the TweetSumm dataset, 25.2 to 26.1 on the WikiHow dataset, and 27.7 to 29.1 on the ArXiv/PubMed dataset. GPT4 relabeling leads to good results even when using just PreSumm’s confidence to score pseudo-labels, suggesting that most GPT4 pseudo-labels are high-quality. The performance after GPT4 relabeling is competitive with the fully supervised PreSumm model trained on 300 labels. This is promising, considering we do not include any examples in the relabeling prompt (making this GPT4’s zero-shot performance). We may further improve the model performance by including some examples in the prompt. Our proposed method outperforms TSL (50:500) and TSL (500:500) despite working in a more challenging labeled:unlabeled dataset ratio of 50:250.

To understand the effectiveness of relabeling, we plot the performance of different semi-supervised settings over multiple iterations of our semi-supervised algorithm (refer to Figure 6). From the curves, we note that the performance of the “Random" setting is highly unstable. Next, while“Conf w/ GPT4 score" improves over the “Conf", they are quite close in the initial iterations. For ArXiv/PubMed, they are almost identical, suggesting the difficulty of dataset. Finally, we note that relabelling not only helps boost performance but also does so at a much faster rate. We note that for all the datasets, our proposed strategy utilizing GPT4 relabeling peaks and stabilizes under 20 iterations (which translates to just 100 new examples).

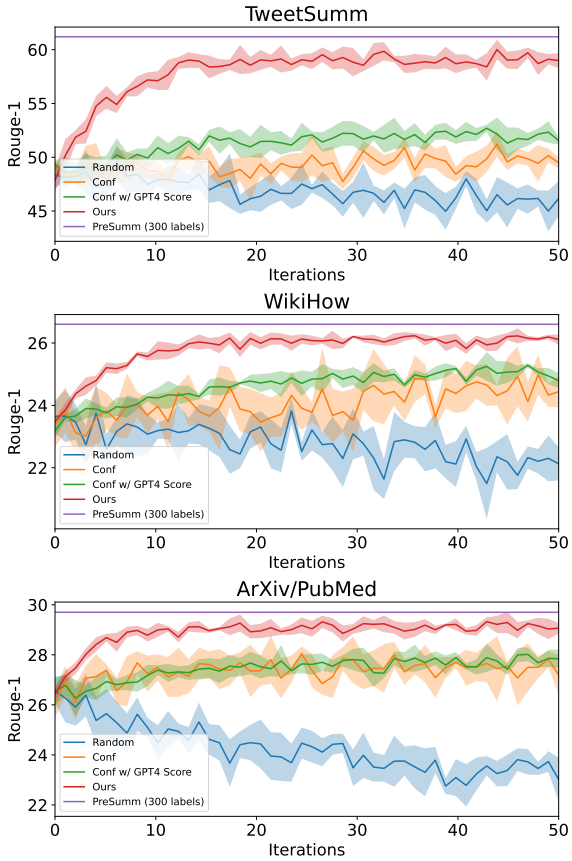


Figure 6: **ROUGE-1 curves.** The plots here show the improvements in ROUGE-1 scores over multiple iterations. **Note:** All results for BERT$_{base}$ backbone for PreSumm.

## 5 Conclusion and Future Work

This work proposes a novel strategy for selecting pseudo-labels to enhance semi-supervised approaches for low-resource text summarization. We show that adding a GPT4 critic on top of a teacher’s confidence leads to the selection of better quality pseudo-labels. Critically, pseudo-labeling with GPT4 leads to significant performance gains, making the use of LLMs like GPT4 for semi-supervised approaches highly promising.

We also identify further improvements that can be made to our approach. For instance, we relabel all the top 50 examples selected by PreSumm. This is cheaper than pseudo-labeling every unlabeled datapoint, but there’s a possibility that GPT-4 relabels an already good-quality summary generated by PreSumm. Therefore, we need better evaluation strategies to decide whether we should relabel an unlabeled datapoint in the first place. Such a method would further reduce costs for using closed-source models like GPT-4. Alternatively, we may

also explore the feasibility of open-source models for this task. We hope this work furthers research toward incorporating LLMs to address such problems for low-resource semi-supervised tasks.

## Limitations

Our results indicate a promising potential to use LLMs for semi-supervised text summarization; however, this work has a few limitations, as detailed in this section. First, we rely completely on GPT's pseudolabels for scoring and relabelling, which is not perfect and we do not currently have a way to automatically judge them. We need a human in the loop to judge the quality of such pseudolabels. GPT-4 is also a closed-source model, and the performance of open-source models for the task is worth noting.

## Ethics Statement

We use GPT to generate new examples, and even though GPT4 is instruction-tuned, some generations might depict undesirable biases with respect to the current objective. For the stated reason, we recommend using our proposed classification models with human monitoring to avoid any ethical issues due to false positive predictions of the downstream classifiers. Practitioners may also consider explicitly debiasing language models for their specific use cases (Barikeri et al., 2021; Schick et al., 2021).

## References


Massih-Reza Amini and Patrick Gallinari. 2001. Automatic text summarization using unsupervised and semi-supervised learning. In *Principles of Data Mining and Knowledge Discovery: 5th European Conference, PKDD 2001, Freiburg, Germany, September 3–5, 2001 Proceedings 5*, pages 16–28. Springer.

Soumya Barikeri, Anne Lauscher, Ivan Vulić, and Goran Glavaš. 2021. Redditbias: A real-world resource for bias evaluation and debiasing of conversational language models. *arXiv preprint arXiv:2106.03521*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jian, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D Hwang, et al. 2023. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*.

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*, pages 245–260, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Wenjie Li, Mingli Wu, Qin Lu, Wei Xu, and Chunfa Yuan. 2006. Extractive summarization using inter- and intra-event relevance. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 369–376.

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.

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)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Nishant Mishra, Gaurav Sahu, Iacer Calixto, Ameen Abu-Hanna, and Issam H Laradji. 2023. Llm aided semi-supervision for extractive dialog summarization. *arXiv preprint arXiv:2311.11462*.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Çelebi, Stanko Dimitrov, Elliott Franco Drabek, Ali Hakim, Wai Lam, Danyu Liu, et al. 2004. Mead-a platform for multidocument multilingual text summarization. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*.

Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 47–57, Dublin, Ireland. Association for Computational Linguistics.

Gaurav Sahu, Olga Vechtomova, Dzmitry Bahdanau, and Issam H Laradji. 2023. Promptmix: A class boundary augmentation method for large language model distillation. *arXiv preprint arXiv:2310.14192*.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *Transactions of the Association for Computational Linguistics*, 9:1408–1424.

Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd international conference on computational linguistics (Coling 2008)*, pages 985–992.

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*, pages 1069–1076.