Dataset Distillation with Attention Labels for Fine-tuning BERT

Aru Maekawa, Naoki Kobayashi, Kotaro Funakoshi, and Manabu Okumura
Tokyo Institute of Technology
{maekawa, kobayasi, funakoshi, oku}@lr.pi.titech.ac.jp

Abstract

Dataset distillation aims to create a small dataset of informative synthetic samples to rapidly train neural networks that retain the performance of the original dataset. In this paper, we focus on constructing distilled few-shot datasets for natural language processing (NLP) tasks to fine-tune pre-trained transformers. Specifically, we propose to introduce attention labels, which can efficiently distill the knowledge from the original dataset and transfer it to the transformer models via attention probabilities. We evaluated our dataset distillation methods in four various NLP tasks and demonstrated that it is possible to create distilled few-shot datasets with the attention labels, yielding impressive performances for fine-tuning BERT. Specifically, in AGNews, a four-class news classification task, our distilled few-shot dataset achieved up to 93.2% accuracy, which is 98.5% performance of the original dataset even with only one sample per class and only one gradient step.

1 Introduction

Deep learning models have achieved state-of-the-art performance in various fields, including computer vision and natural language processing (NLP), using large-scale neural networks trained with huge datasets. Unfortunately, their successful performances have come with massive training costs, including training time, GPU resources, and energy consumption. To reduce the training costs, current research has been focusing on constructing a small training dataset such that models trained with it can achieve comparable performances to models trained with the whole original dataset.

One classical way to compress the training dataset is data selection. Data selection methods choose a subset of effective training samples on the basis of a number of heuristic measures, for example, cluster centers (Sener and Savarese, 2018), diversity (Aljundi et al., 2019), and likelihood of models (Moore and Lewis, 2010). Although the data selection methods effectively work for efficient model training and several applications, such as active learning (Sener and Savarese, 2018) and continual learning (Aljundi et al., 2019), their performance is clearly restricted because they rely on the existence of representative samples that are effective for model training in the original dataset.

As an alternative approach for reducing the training dataset, Wang et al. (2018b) proposed *dataset distillation*, which aims to create a small number of synthetic samples optimized to effectively train models. Dataset distillation has attracted much attention in machine learning (Wang et al., 2018b; Zhao et al., 2021; Zhao and Bilen, 2021; Sucholutsky and Schonlau, 2021; Bohdal et al., 2020; Wang et al., 2022; Cazenavette et al., 2022) for both the theoretical interest and various applications, such as neural architecture/hyper-parameter search (Such et al., 2020), continual learning (Masarczyk and Tautkute, 2020; Rosasco et al., 2022), federated learning (Goetz and Tewari, 2020; Zhou et al., 2020), and preserving data privacy (Li et al., 2020; Dong et al., 2022).

However, most of the existing research on dataset distillation mainly focuses on image datasets, and only a few studies involve NLP tasks. Sucholutsky and Schonlau (2021) and Li and Li (2021) extended dataset distillation to text datasets by using embedding vectors as an input of the distilled dataset instead of discrete text. While these studies applied dataset distillation to those model architectures based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs), we cannot find any research that tackles dataset distillation for pre-trained transformers, such as BERT (Devlin et al., 2019), which have become the de-facto standard for various kinds of NLP tasks. Therefore, in this paper, we aim to obtain distilled few-shot datasets to fine-tune the pre-trained transformers for NLP tasks.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 119–127
July 9-14, 2023 ©2023 Association for Computational Linguistics

To this end, we focus on the attention mechanism, which is the core component of transformers (Vaswani et al., 2017). Several current studies utilized supervision of the attention probabilities to effectively train the model (Liu et al., 2016; Mi et al., 2016). Moreover, it is also used for the model distillation to efficiently transfer the knowledge of a transformer model to another one via attention probabilities (Aguilar et al., 2020; Jiao et al., 2020; Sun et al., 2020; Wang et al., 2020, 2021). Inspired by this, we propose distilled attention labels, which are the supervision of attention probabilities optimized as a part of the distilled dataset, to enhance the effectiveness of the distilled dataset for training the transformer models.

In our experiments, we constructed distilled few-shot datasets to fine-tune BERT (Devlin et al., 2019) in various types of NLP tasks: AGNews (text classification), SST-2 (sentiment analysis), QNLI (QA/NLI), and MRPC (paraphrase identification).

Our main contributions are as follows: (i) To the best of our knowledge, this is the first work to explore dataset distillation for pre-trained transformers. Specifically, we demonstrate that our distilled datasets effectively fine-tune BERT even with only one sample for each class and only one gradient step. (ii) We present the distilled attention labels, which can easily be applied to dataset distillation for transformer architectures. Experimental results show that they consistently improved the performance with the distilled datasets in various types of NLP tasks. (iii) We open our source code and the distilled datasets obtained through our experiments to facilitate further research.[1]

2 Methodology

2.1 Dataset Distillation

In this section, we explain the basic approach of dataset distillation (Wang et al., 2018b), which aims to optimize a synthetic dataset through the gradient method similar to the current meta-learning approach (Finn et al., 2017).

Let the original training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where (x_i, y_i) is a pair of an input and its class label. Our goal is to optimize a distilled dataset $\tilde{\mathcal{D}} = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{M}$, which is randomly initialized at first, with $M \ll N$.

The model parameters θ are updated with a mini-batch of the distilled dataset $(\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{y}}_t)$ by gradient descent (GD) steps as follows:

$$\begin{aligned} \theta_{t+1} &= \theta_t - \tilde{\eta} \nabla_{\theta_t} \mathcal{L}_{task} \\ \text{s.t.} \quad \mathcal{L}_{task} &= L(\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{y}}_t, \theta_t), \end{aligned} \tag{1}$$

where $L()$ is a twice-differentiable loss function and $\tilde{\eta}$ is the learnable learning rate of the model, which is optimized together with $\tilde{\mathcal{D}}$. Given initial model parameters θ_0, we can represent the model trained with the distilled dataset $\tilde{\mathcal{D}}$, with the number of GD steps T, as

$$\theta_T = F(\theta_0; \tilde{\mathcal{D}}, \tilde{\eta}, T), \tag{2}$$

where $F()$ is the training procedure of the T steps for the GD updating (Eq. 1).

As the goal of dataset distillation is that θ_T performs well on the original dataset, the optimization objective of the distilled dataset $\tilde{\mathcal{D}}$ is calculated as follows:

$$\begin{aligned} \mathcal{L}_{distill}(\tilde{\mathcal{D}}, \tilde{\eta}; \theta_0) &\coloneqq L(\boldsymbol{x}_t, \boldsymbol{y}_t, \theta_T) && (3) \\ &= L(\boldsymbol{x}_t, \boldsymbol{y}_t, F(\theta_0; \tilde{\mathcal{D}}, \tilde{\eta}, T)), && (4) \end{aligned}$$

where $(\boldsymbol{x}_t, \boldsymbol{y}_t)$ is a mini-batch of the original training dataset.

Therefore, the optimization problem for dataset distillation is formulated as

$$\tilde{\mathcal{D}}^*, \tilde{\eta}^* = \arg\min_{\tilde{\mathcal{D}}, \tilde{\eta}} \mathbb{E}_{\theta_0 \sim p(\theta_0)} \left[\mathcal{L}_{distill}(\tilde{\mathcal{D}}, \tilde{\eta}; \theta_0) \right], \tag{5}$$

where $p(\theta_0)$ is the distribution of θ_0.

We optimize the distilled dataset $\tilde{\mathcal{D}}$ with this objective by using current gradient-based optimization techniques, e.g., Adam (Kingma and Ba, 2015). However, the discrete nature of text data makes it difficult to apply the gradient methods directly. Inspired by previous work (Sucholutsky and Schonlau, 2021; Li and Li, 2021), we use a sequence of embedding vectors for inputs of the distilled dataset instead of text as it is. Using the embeddings makes the loss $\mathcal{L}_{distill}$ differentiable with respect to $\tilde{\mathcal{D}}$, and we can thus optimize the distilled dataset $\tilde{\mathcal{D}}$ by the gradient methods.

2.2 Distilled Soft Labels

The class labels of the original dataset are usually discrete hard labels (i.e., one-hot labels representing only a single class). Instead of hard labels, we can use soft labels for distilled datasets and optimize them with the input embeddings. Using soft

[1] https://github.com/arumaekawa/dataset-distillation-with-attention-labels

labels enables the distilled datasets to contain more information. Following previous work (Sucholutsky and Schonlau, 2021; Bohdal et al., 2020), we first initialize the soft labels with one-hot values and enable them to take any real values. We can now optimize the soft labels through the gradient method as well as the input embeddings.

2.3 Distilled Attention Labels

For efficient knowledge transfer to transformer models via training with the distilled dataset, we propose attention labels, which are optimized to guide the multi-head attention module of the transformer models.

Inspired by previous work (Aguilar et al., 2020; Wang et al., 2020, 2021), we compute the Kullback-Leibler (KL) divergence D_{KL} between the self-attention probabilities of the model $a(\theta)$ and the distilled attention labels $\tilde{a}$ across all layers and heads. The attention loss $\mathcal{L}_{attn}$ is computed as follows:

$$\mathcal{L}_{attn} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{H}\sum_{h=1}^{H} D_{\mathrm{KL}}(\tilde{a}_{k,h}||a_{k,h}(\theta)), \quad (6)$$

where $\tilde{a}_{k,h}$ and $a_{k,h}(\theta)$ are the attention maps for the h-th head of the k-th layer of the distilled attention labels and the model, respectively, K is the number of layers, and H is the number of heads. Due to the data size, we consider the attention probabilities only for the first input token (`[CLS]`).

We train the model to minimize $\mathcal{L}_{task}$ and $\mathcal{L}_{attn}$ at the same time. Thus, the GD updating of the model (Eq. 1) is modified as

$$\theta_{t+1} = \theta_t - \tilde{\eta}\nabla_{\theta_t}(\mathcal{L}_{task} + \lambda\mathcal{L}_{attn}), \quad (7)$$

where λ is the balance weight for $\mathcal{L}_{attn}$.

The attention labels $\tilde{a}$ are first initialized randomly and restricted to being a valid probability distribution (i.e., non-negative and the sum equals 1) by applying the softmax function to real-valued vectors. We optimize the attention labels together with the input embeddings and the soft labels by the gradient method. The details of the step-by-step procedure of our distillation algorithm are shown in Appendix A.

3 Experiments

3.1 Settings

Datasets. We evaluated our dataset distillation methods in various types of NLP tasks. We used a text classification task (AGNews (Zhang et al., 2015)) and three different natural language understanding tasks (SST-2, QNLI, and MRPC) from the GLUE benchmark (Wang et al., 2018a). For the evaluation metrics, we used accuracy for AGNews. For the other three tasks, we followed the evaluation settings of GLUE (Wang et al., 2018a). The statistics of each benchmark dataset are summarized in Table 1.

Dataset	Task	Metric	C	# Train	# Test (Dev.)
AGNews	news classification	acc.	4	120k	7.6k
SST-2	sentiment	acc.	2	67k	872
QNLI	QA/NLI	acc.	2	105k	5.5k
MRPC	paraphrase	acc./F1	2	3.7k	408

Table 1: Summary of evaluation datasets. C represents the number of classes in each task. For the three GLUE tasks, since the test set is not available, we report the evaluation results on the development set.

Network Architecture. To evaluate the dataset distillation methods, we constructed distilled few-shot datasets to fine-tune BERT (Devlin et al., 2019), which is the first pre-trained transformer model, that all subsequent models are based on. We utilized the pre-trained $\mathrm{BERT}_{\mathrm{BASE}}$ model. Following the fine-tuning procedure in Devlin et al. (2019), we introduced additional classification layer weights $W \in \mathbb{R}^{C\times D}$ on the last hidden state of the `[CLS]` token, where D is the hidden dimension of BERT and C is the number of classes.

Implementation. For all our distilled datasets, we used Adam optimizer (Kingma and Ba, 2015) with a learning rate $\alpha \in \{1e^{-3}, 1e^{-2}, 1e^{-1}\}$ and trained the distilled datasets for 30 epochs. We initialized the learnable learning rate $\tilde{\eta} \in \{1e^{-2}, 1e^{-1}\}$. For the attention labels, we set $\lambda = 1.0$, which performed well in our preliminary experiments. We report the results for the best performing combination of α and $\tilde{\eta}$. Note that due to the coarse granularity of the search, there is no need to care about overfitting to the test set. More details of our implementation are shown in Appendix B.

Evaluation. To evaluate the distilled datasets, we fine-tuned the BERT model with them for 100 times, where the additional parameters W were randomly initialized each time. In all our experiments, we report the mean and standard deviation over the 100 evaluation results.

3.2 Results for 1-shot and 1-step Setting

We first evaluated the dataset distillation methods with a 1-shot and 1-step setting, where the distilled

	AGNews	SST-2	QNLI	MRPC
Majority	25.0	50.9	50.5	74.8
HL	87.4±1.8	81.6±2.4	68.6±2.5	74.8±0.0
SL	88.4±0.9	82.5±1.6	76.4±0.8	74.8±0.0
HL + AL	**93.2±0.1**	**90.1±0.3**	85.9±0.1	76.4±0.8
SL + AL	93.0±0.1	89.0±0.2	**86.4±0.1**	**78.8±0.7**
Full dataset	94.6	92.7*	91.8*	88.6*

Table 2: Experimental results for the 1-shot and 1-step setting. 'HL' and 'SL' mean hard and soft class labels, respectively, and 'AL' means attention labels. 'Majority' is the majority class baseline. Scores for the full dataset with * are cited from Devlin et al. (2019). Bold scores show the best results for each task.

# step	# shot	AGNews	SST-2	QNLI	MRPC
Single-step setting					
1	1	93.0±0.1	89.0±0.2	86.4±0.1	78.8±0.7
1	3	**93.5±0.1**	90.3±0.2	86.7±0.1	79.3±0.5
1	5	93.1±0.1	90.1±0.2	86.9±0.1	79.4±0.5
Same distilled data for each step					
3	1	93.0±0.1	89.8±0.4	84.2±0.4	74.8±0.0
5	1	92.1±0.1	85.8±0.4	85.9±0.1	74.8±0.0
Different distilled data for each step					
3	3	92.5±0.1	90.4±0.2	**87.0±0.1**	**80.3±0.8**
5	5	93.1±0.1	**90.7±0.2**	86.1±0.1	76.5±0.8

Table 3: Experimental results for the multiple-shot and multiple-step setting. Bold scores show the best results for each task.

dataset includes only one sample per class, and BERT was fine-tuned with it by only one GD step. We compared the performance for hard/soft labels and with/without attention labels for each task.

Table 2 shows the evaluation results. The distilled datasets with the hard labels, i.e., only optimizing the input embeddings and not applying the attention labels, still achieved 87.4, 81.6, and 68.6 for AGNews, SST-2, and QNLI, respectively, which is 92.4, 88.0, and 74.7% performance of the full dataset. Furthermore, using the soft labels further improved these performances, especially by almost 8 points for QNLI. However, for MRPC, the distilled dataset achieved only the same performance as the majority class baseline regardless of the use of the soft labels.

When applying the attention labels, the performance of the distilled dataset was significantly improved for all tasks, and their effect is much greater than the soft labels. Specifically, our distilled dataset with the attention labels yielded up to 98.5, 97.2, 94.1, and 88.9% performance of the full dataset for AGNews, SST-2, QNLI, and MRPC, respectively. These results indicate that using the attention labels enables to extract the information from the original dataset as the attention probabilities and to efficiently transfer it to the model.

When comparing the performance between the four tasks, dataset distillation performed very well on relatively simple classification tasks such as AGNews and SST-2, while the performance was somewhat limited on QNLI and MRPC, which require understanding the relationship between two sentences. In particular, for MRPC, although the performance was improved by applying the attention labels, the gap from the full dataset was still larger than that in the other three tasks. The class imbalance in the original training dataset (68% positive) may make the training of the distilled dataset more difficult. We can say there is still room for performance improvement by dealing with this issue (e.g., by upsampling or downsampling).

3.3 Results for Multiple-shot and Multiple-step Setting

We also evaluated the distilled datasets with more than one shot and more than one GD step to fine-tune BERT. For the multiple-step setting, we considered two different scenarios: using the same distilled data in all steps and using different distilled data for each step. In these experiments, we evaluated the distilled datasets that use soft labels and attention labels for different numbers of GD steps $T \in \{1, 3, 5\}$.

Table 3 shows the results for the multiple-shot and multiple-step setting. In the single-step setting, overall performance improved with the number of shots of the distilled data. We believe that this is simply due to the expressiveness of the distilled data improved with the size of them. When using the same distilled data for all steps in the multiple-step setting, the performance of the distilled datasets degraded even compared with that in the single-step setting. In contrast, the performance was improved by separating the distilled data for each step and slightly but better than that with the same number of shots in the single-step setting. These results suggest that the role of the distilled data is different between the earlier and later steps, and it is difficult to obtain the distilled data that are generally useful for all GD steps.

In addition, the basic dataset distillation algorithm we used requires computing the back propagation through all GD steps for the optimization of

the distilled dataset, which increases memory and computational costs linearly with T. Therefore, it was difficult to increase T to be larger than 5 in our experiments. This is the limitation of our dataset distillation method, and it needs further improvement to scale to more complex tasks or to train models from scratch.

4 Conclusion

In this paper, we explored dataset distillation in NLP tasks to fine-tune pre-trained transformers. We proposed attention labels, which are the supervision of attention probabilities distilled as a part of the distilled datasets. Experimental results across various tasks demonstrate that our distilled few-shot datasets achieved successful performances even with only one sample per class. Notably, the attention labels significantly improved the performance of the distilled datasets even for the tasks where dataset distillation is difficult without them.

Limitations

We think the following three points are the limitations of this work. (i) As mentioned in Section 3.3, the computational cost of our distillation approach increases linearly with the number of GD steps and the distilled data size. It is necessary to explore efficient distillation algorithms to scale our method to more complex tasks or full-scratch training in future work. (ii) To optimize the distilled dataset through the gradient method, we utilized word embedding vectors instead of directly optimizing the text as in the existing work. Therefore, the distilled dataset we obtained cannot be applied to models with different word embeddings, such as other pre-trained models or full-scratch training. (iii) In our experiments, we evaluated our approach only on text classification tasks. However, our approach can also be applied to text generation tasks as well by applying the attention labels to all input tokens (not only [CLS]) and using vocabulary-wise soft labels. In future work, we should investigate its performance and explore more effective approaches.

References

Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. 2020. Knowledge distillation from internal representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7350–7357.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. 2019. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ondrej Bohdal, Yongxin Yang, and Timothy M. Hospedales. 2020. Flexible dataset distillation: Learn labels instead of images. *CoRR*, abs/2006.08572.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. 2022. Dataset distillation by matching training trajectories. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*, pages 4749–4758. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tian Dong, Bo Zhao, and Lingjuan Lyu. 2022. Privacy for free: How does dataset condensation help privacy? In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5378–5396. PMLR.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.

Jack Goetz and Ambuj Tewari. 2020. Federated learning via synthetic data. *CoRR*, abs/2008.04489.

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 *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. 2020. Soft-label anonymous gastric x-ray image distillation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 305–309.

Yongqi Li and Wenjie Li. 2021. Data distillation for text classification. *CoRR*, abs/2104.08448.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan. The COLING 2016 Organizing Committee.

Wojciech Masarczyk and Ivona Tautkute. 2020. Reducing catastrophic forgetting with learning on synthetic data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pages 1019–1024. Computer Vision Foundation / IEEE.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2283–2288, Austin, Texas. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Andrea Rosasco, Antonio Carta, Andrea Cossu, Vincenzo Lomonaco, and Davide Bacciu. 2022. Distilled replay: Overcoming forgetting through synthetic samples. In *Continual Semi-Supervised Learning*, pages 104–117, Cham. Springer International Publishing.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth Stanley, and Jeffrey Clune. 2020. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9206–9216. PMLR.

Ilia Sucholutsky and Matthias Schonlau. 2021. Soft-label dataset distillation and text dataset distillation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. 2022. Cafe: Learning to condense dataset by aligning features. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12186–12195.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. 2018b. Dataset distillation. *CoRR*, abs/1811.10959.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Bo Zhao and Hakan Bilen. 2021. Dataset condensation with differentiable siamese augmentation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12674–12685. PMLR.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2021. Dataset condensation with gradient matching. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. 2020. Distilled one-shot federated learning. *CoRR*, abs/2009.07999.

Algorithm 1 Dataset Distillation with Attention Labels

Input: Training dataset D, distribution of initial parameters $p(\theta_0)$, number of outer-loop steps S, number of inner-loop steps T, initial learnable learning rate $\tilde{\eta}_0$, learning rate for the distilled dataset α, balanced weight for the attention loss λ.

Initialize distilled dataset: $\tilde{\mathcal{D}} = \{(\tilde{x}_i, \tilde{y}_i, \tilde{a}_i)\}_{i=1}^{M}$ randomly
Initialize learnable learning rate: $\tilde{\eta} \leftarrow \tilde{\eta}_0$
for outer step $s = 1, \ldots, S$ **do**
    Initialize parameters: $\theta_0 \sim p(\theta_0)$
    for inner step $t = 1, \ldots, T$ **do**
        Get the t-th mini-batch of distilled data:
            $(\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{y}}_t) \sim \tilde{\mathcal{D}}$
        Compute task loss $\mathcal{L}_{task} = L(\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{y}}_t, \theta_{t-1})$
        Compute attention loss $\mathcal{L}_{attn}$ flowing Eq. 6
        Update parameters:
            $\theta_{t+1} = \theta_t - \tilde{\eta} \nabla_{\theta_t} (\mathcal{L}_{task} + \lambda \mathcal{L}_{attn})$
    end for
    Sample a mini-batch of real data: $(\boldsymbol{x}_s, \boldsymbol{y}_s) \sim \mathcal{D}$
    Update distilled data:
        $\tilde{\mathcal{D}} \leftarrow \tilde{\mathcal{D}} - \alpha \nabla_{\tilde{\mathcal{D}}} L(\boldsymbol{x}_s, \boldsymbol{y}_s, \theta_T)$
end for

Output: Distilled dataset $\tilde{\mathcal{D}}$ and learning rate $\tilde{\eta}$

A Overview of Proposed Method

Algorithm 1 illustrates an overview of our distillation algorithm.

B Implementation details

In our experiments, we trained the distilled datasets using Adam optimizer (Kingma and Ba, 2015) with linear warmup and linear decay learning rate schedule and gradient clipping with 1.0. Following the implementation in Wang et al. (2018b), we disabled dropout layers to avoid the randomness of the model training. We used a RTX 3090 or a RTX A6000, depending on the required memory size for each experiments. To obtain the performance of the full dataset for AGNews, which is used as the upper-bound of the distilled datasets, we fine-tuned $\text{BERT}_{\text{BASE}}$ model with learning rate $\eta = 1e^{-5}$ for epochs $\in \{2, 3, 4\}$, and adopted the best performance. More information about our implementation can be found in our source code[1].

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.