Improving MetaDistil Framework

Abhinav Lugun **Anh Nguyen** **Maria Rahaman Rumki**
st122322@ait.asia 122910@ait.asia st123835@ait.asia
School of Engineering and Technology, Asian Institute of Technology, Bangkok, Thailand

1. Introduction

Large models have gained significant popularity due to their ability to enhance machine learning performance. However, their deployment is not ideal, given the strict latency and computational resource limitations. To address this challenge, many techniques have been developed, including knowledge distillation (Hinton et al, 2015), which involves transferring knowledge from a larger teacher model to a smaller student model. While this technique has been proven effective in computer vision and natural language tasks, there is still room for improvement. Zhou et al. (2022) highlighted two limitations of the conventional knowledge distillation method which are (1) the teacher model's lack of awareness of the student model's capacity and (2) the teacher model not optimized to distill knowledge effectively.

To mitigate these issues, Zhou et al. (2022) proposed Knowledge Distillation with Meta-Learning (MetaDistil) framework as an alternative to conventional knowledge distillation. MetaDistil framework is an advanced approach to knowledge distillation that has shown promising results in improving the knowledge distillation process of large-scale machine learning models to smaller ones. While conventional knowledge distillation techniques have been successful in compressing models for various natural language and computer vision tasks, they still have limitations that would need to be addressed. This framework uses the concept of meta-learning to allow the teacher model to learn how to teach or pass knowledge more efficiently by getting feedback from the student model during the distillation process. In this way, the teacher model improves its knowledge transfer ability during the distillation process. Through meta-learning, the MetaDistil framework allows the teacher model to consider the student model's capacity during the distillation process, optimizing knowledge transfer and resulting in more effective teaching and improved efficiency (Tian et al., 2022). (Pan et al., 2021) The Meta-KD framework shows promising results against several advanced techniques, such as conventional knowledge distillation and model fine-tuning in terms of student model performance.

In this work, we do a few experiments which we hope will serve as an improvement to the MetaDistil framework. In one of the experiments, the student in addition to learning from one teacher model will also learn from another at the same time. Theoretically, learning from a single teacher model may only provide limited or biased knowledge (Wu et al., 2021). Also, having more than one teacher helps to deal with random seed dependency problems to a certain extent (Ilichev et al., 2021).

In another experiment, we train the student model partially before doing the distillation process. Generally, the student model in the MetaDistil framework is yet to be trained. However, we wish to test that training the student on the task on which the teacher model is trained will help the student model to better learn as it has some familiarity with the knowledge it will be learning. This is inspired by the analogy that the student has a better understanding of the lecture being taught by the teacher if he/she goes through the class material before the lecture.

Finally, we experiment with injecting noise in both the teacher and the student model. Liu et al, (2021) point out training with noises will make the model more robust.

By introducing noise, learning from an additional teacher, and training the student model partially beforehand, we expect the student model to have a better ability to absorb knowledge from the teacher model while also being more robust to inherent dataset noises. This hopefully would result in improved performance on NLP tasks. This would be tested out in the summarization task which should have a similar effect, though not the same rate, on other tasks.

In summary, our contributions to the existing MetaDistil are:

- Student model learning from one additional teacher also
- Training the student model partially
- Conducting experimenting with noise on both teacher and student model

2. Related Work

2.1 Knowledge Distillation

Knowledge Distillation (Bucila et al., 2006; Hinton et al., 2015; Gou et al., 2022) is widely used in natural language processing as a technique for transferring knowledge from a larger and more complex model (teacher) to a smaller and simpler model (student). The approach was first introduced by Hinton and his colleagues, who demonstrated that incorporating the soft label generated by the teacher model along with the correct label improved the performance of the student model, bringing it closer to that of the teacher model. Since then, researchers have explored various approaches to enhance the effectiveness of knowledge distillation in NLP. There are various approaches to knowledge distillation in deep learning, as documented in the literature. Some of these methods include utilizing different latent representations, such as those proposed by (Romero et al., 2015, Zagoruyko & Komodakis 2017, Tung & Mori 2019, Park et al., 2019, Sun et al., 2019, and Jiao et al., 2019). Other methods involve using multiple teacher models together, as done by (You et al., 2017; Liu et al., 2021), or updating both teacher and student models simultaneously with added constraints, as proposed by Meng et al. (2019). Another approach is deep mutual learning, where multiple models collaborate and teach each other, as demonstrated by (Zhang et al., 2018; Zhao et al., 2021).

Additionally, some researchers have adopted a patient learning mechanism that extracts knowledge from previous layers of the teacher network instead of only the last layer (Siqi Sun et al., 2019), while Sergey et al. (2017) introduced attention mechanisms to enhance the performance of the student model. FitNet aimed to mimic the teacher model by transferring intermediate representations to the student model (Adriana et al., 2015) . Baoyun et al. (2019) focused on the correlation between instances to transfer, while ReviewKD proposed a review mechanism to transfer multi-level knowledge from the teacher to the student (Hengshuang et al., 2021). Finally, some researchers have implemented multi-step knowledge distillation, which involves an intermediate-sized network (teacher assistant) to bridge the gap between the student and the teacher (Seyed Iman et al., 2019).

Overall, these methods aim to extract more informative knowledge from the teacher model and transfer it effectively to the student model, resulting in better performance on various NLP tasks

2.2 Meta Learning

Meta-learning is a rapidly evolving field that aims to create algorithms that can improve the learning process. (Bansal et al., 2021; Lee et al., 2021; Chen et al., 2020) are some recent studies in this area. The core concept of meta-learning is "learning to learn," where an algorithm can adjust its learning strategy based on feedback from the environment (Tian et al., 2022). Meta-learning typically involves two levels of learning: the meta-level and the task-level (Antoniou et al., 2019).

At the meta-level, a "meta-learner" optimizes its learning algorithm based on the feedback it receives from the "inner-learner" on specific tasks. Some recent works have combined knowledge distillation with meta-learning. Finn et al. (2019) introduce an online meta-learning framework that allows a model to continually adapt and learn new tasks online, while Li et al. (2017) propose a method for knowledge distillation that transfers the selective behavior of neurons from the teacher model to the student model.

Jabri et al. (2019) investigated unsupervised task generation in meta-reinforcement learning based on different variations of a reward function. (Kaddour et al., 2020; R Luna Gutierrez and M Leonetti., 2020) proposed a task sampling strategy from existing meta-training tasks, where the probability of selecting a task is proportional to the amount of information it provides.

In addition to these methods that are specific to reinforcement learning, there have also been proposals to address supervised meta-learning problems. For example, Li et al. (2020) proposed a difficulty-aware meta-loss function that assigns higher weights to harder samples during meta-training to improve generalization to new tasks. Liu et al. (2020) proposed a greedy class-pair based task sampling strategy to enhance model adaptation to new tasks by selecting tasks that are semantically related. The various approaches and methods proposed in meta-learning aim to improve the efficiency and effectiveness of the learning process, enhance model generalization and adaptation to new tasks, and reduce the need for extensive training on new tasks.

2.3 Knowledge Distillation with Meta Learning

Much work has been done on knowledge distillation using meta-learning during the last few years. For example, (Pan et al., 2020) incorporate it for the teacher model to absorb more different domains to improve its generality capacity. It involves two learning models, the meta-model is trained using a set of meta-tasks while the routing network is trained to transfer the knowledge from the teacher model to the meta model. Another instance, from which the experiment is further carried on, is from (Zhou et al., 2022) where meta learning helps the teacher model to improve its knowledge transfer based on student model feedback. Pan et al. (2021) propose a novel method called "Meta-KD," which combines meta-learning and knowledge distillation to improve the performance and generalization of the student model. The method involves learning a task-agnostic representation of the input data, which can be used to facilitate knowledge transfer across different tasks. The authors show that their method outperforms existing approaches on several benchmarks. Huang et al. (2021) experiments using a small dataset to learn a meta-model that can quickly adapt to new tasks. The method entails learning a set of auxiliary tasks, using them to train a set of base models, and then using a ‘dense interaction approach’ to distill those base models into meta models.

3. Methodology

We use the MetaDistil framework done in (Zhou et al., 2022). In addition, we experiment with **(1)** student learning from an additional teacher model, **(2)** training the student model beforehand partially on the task on which it will be absorbing knowledge from the teacher model and **(3)** introducing noise to the dataset.

Overview of MetaDistil framework

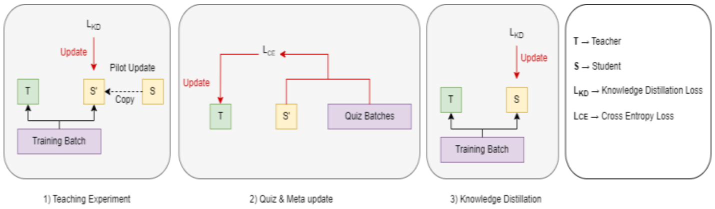


Figure 1: Workflow of MetaDistil from (Zhou et al., 2022)

In contrast to conventional knowledge distillation, the teacher model is trainable during the distillation process so that the teacher model can adjust itself through the whole process for a better transfer of knowledge. It is assumed that the teacher model is already trained and has high accuracy for a particular task or multiple tasks.

Figure (1) shows the workflow of the MetaDistil framework from (Zhou et al., 2022). The whole process can be broken down into three major steps:

- **(1) Teacher experiment:** During the distillation process, a copy of the student model is made. Then, the update is made to the parameters of the copy student according to the distillation loss. This, the teacher experiments the distillation process on the copy student model on a certain sample of data before updating the parameters of the original student model.
- **(2) Quiz & Meta Update:** The copy student is tested on the quiz sample set aside from the dataset to test its progress by calculating its cross-entropy loss. This progress is given as feedback ("by calculating second derivatives and performing gradient descent") to the teacher model which then uses this feedback to improve its "teaching ability". Basically, The gradients of the cross-entropy loss with respect to the teacher model are calculated and an update/meta-update to the teacher model is made by these calculated gradients.
- **(3) Knowledge Distillation:** Finally, a normal distillation process is followed where the parameters of the original student are updated according to the distillation loss.

The addition of the pilot mechanism to the MetaDistil was one of the important features of the framework. It refers to the process where both teacher and copy student models are updated on a sample of data after which the update to the original student model is made. This mechanism was introduced as meta-learning aims to optimize the meta-learner (teacher model) instead of the inner-learner (student model). By this mechanism, the focus shifts to the optimization of the inner-learner instead by allowing the teacher model, already updated from the feedback, to have the "instant effect" on the student model on the same sample of data.

Modifications

In the MetaDistil framework, the student model would generally learn from one teacher model. However, we introduce an additional teacher model for training the student. Using an extra teacher would provide extra knowledge for the student model. The workflow of MetaDistil would still be the same except that the update is made to both teacher models simultaneously during quiz on the student model. The knowledge from both teacher models are incorporated by averaging their output logits before feeding into softmax function.

In addition, we train the student model partially by doing fine-tuning for one epoch before doing the distillation process. We wish to test the hypothesis that the student model will better absorb knowledge from the teacher model if it has some familiarity with the knowledge beforehand.

Finally, we add noise for both the teacher and the student model. Liu et al. (2021) found injecting noise in both teacher and student models to improve student's performance. One strong reason behind this could be that since data inherently contains noise, training with noise will make the model more robust. Their work on noise configuration settings will be adopted in this experimentation. Details of configurations of noise injection in both teacher and student model is given below.

Noise Injection

In the methodology section, we employed a technique called Noise Injection as part of our

Natural Language Processing (NLP) pipeline Liu et al. (2021). Our methodology incorporates the Noise Injection technique as a fundamental component of our Natural Language Processing (NLP) pipeline. Noise Injection is employed to introduce random noise into sentences or text, with the aim of generating more diverse data and enabling our machine learning models to better understand variations in real-world language (Ko et al., 2020). To implement Noise Injection, we follow a systematic approach. Initially, we tokenize the input sentences into words or subword units. Next, for each word in the sentence, we assign a random value between 0 and 1. If this value is less than the predefined probability p, the word is replaced with a noise word. This process is repeated for each word in the sentence, ensuring a stochastic substitution based on the specified probability. For example, consider the sentence "The cat is sleeping peacefully." with a probability p set to 0.2 (20%). When applying Noise Injection, the word "peacefully" might be replaced with a noise word, resulting in the sentence "The cat is sleeping noise_word." This substitution introduces noise into the sentence, allowing our models to learn from variations in language patterns and improve their adaptability to real-world scenarios.The selection of the noise word can vary depending on the specific requirements of our study. It can be a predefined noise token or a randomly generated word from a noise vocabulary. The choice of noise word aims to introduce variability while maintaining the overall coherence and semantic meaning of the sentence.

4. Experiment

4.1 Experiment Setup

For the experiment, the MetaDistil framework was tested on sentiment analysis tasks. SST-2 dataset was used for the task evaluation. From the dataset, 30,000 samples were set aside for the train dataset, 100 samples for the quiz dataset, 872 samples for validation, and 2,000 for test dataset. Accuracy was used as the reporting metric.

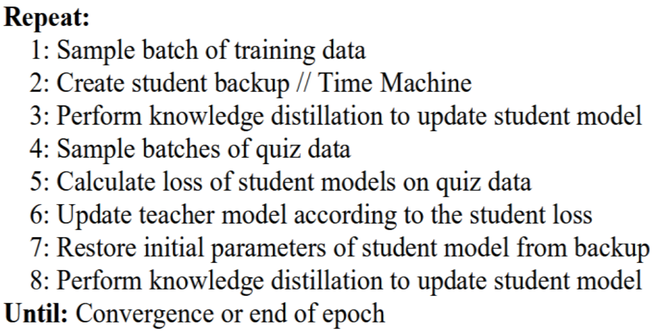
Model

BERT and RoBERT were used as teacher models for training the model. For the student model, TinyBert and DistilBERT were used.

Baselines

For baselines, normal knowledge distillation and MetaDistil was used. Also, a version of models TinyBERT and DistilBERT was directly fine-tuned instead of being trained with the distillation algorithm for baseline consideration These would be compared with modified MetadDistil framework which considers an additional teacher, fine-tuning student model for 1 epoch beforehand, and noise introduction.

Training Details

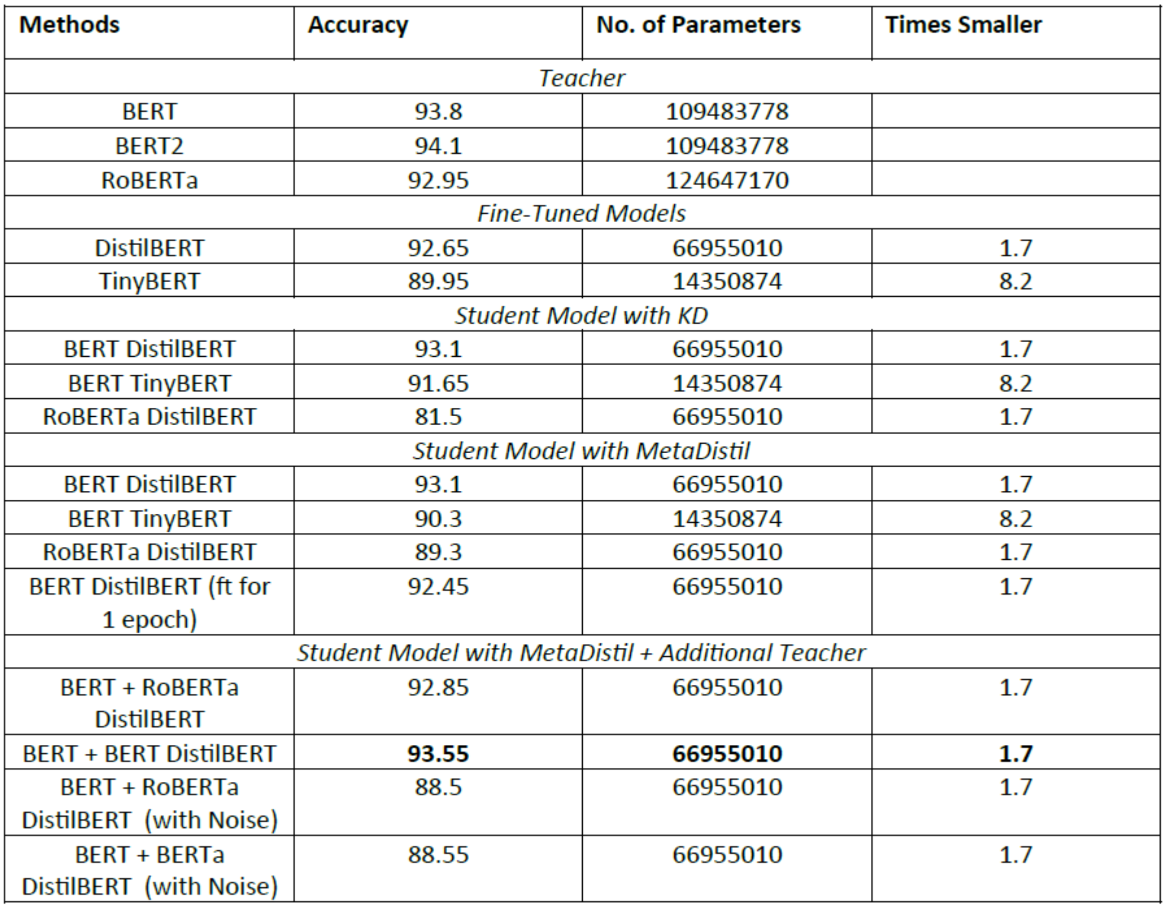


```
Repeat:
  1: Sample batch of training data
  2: Create student backup // Time Machine
  3: Perform knowledge distillation to update student model
  4: Sample batches of quiz data
  5: Calculate loss of student models on quiz data
  6: Update teacher model according to the student loss
  7: Restore initial parameters of student model from backup
  8: Perform knowledge distillation to update student model
Until: Convergence or end of epoch
```

Figure 2: Sudo Code

For the training settings, AdamW optimizer with the learning of 6e-5 was used for both teacher and student models. For the knowledge distillation loss, KL divergence was used. Figure 2 shows the sudo code on which the MetaDistil framework was

Table 1. Experimental Results



Methods	Accuracy	No. of Parameters	Times Smaller
Teacher			
BERT	93.8	109483778	
BERT2	94.1	109483778	
RoBERTa	92.95	124647170	
Fine-Tuned Models			
DistilBERT	92.65	66955010	1.7
TinyBERT	89.95	14350874	8.2
Student Model with KD			
BERT DistilBERT	93.1	66955010	1.7
BERT TinyBERT	91.65	14350874	8.2
RoBERTa DistilBERT	81.5	66955010	1.7
Student Model with MetaDistil			
BERT DistilBERT	93.1	66955010	1.7
BERT TinyBERT	90.3	14350874	8.2
RoBERTa DistilBERT	89.3	66955010	1.7
BERT DistilBERT (ft for 1 epoch)	92.45	66955010	1.7
Student Model with MetaDistil + Additional Teacher			
BERT + RoBERTa DistilBERT	92.85	66955010	1.7
BERT + BERT DistilBERT	**93.55**	**66955010**	**1.7**
BERT + RoBERTa DistilBERT (with Noise)	88.5	66955010	1.7
BERT + BERTa DistilBERT (with Noise)	88.55	66955010	1.7

implemented. Notice that instead of creating a copy model, the model's weights are copied as a backup for conserving GPU memory during training. In the case of the additional teacher incorporation, the step would still be the same except that average of teacher output logits is done on steps 3 and 8, and an update is made to both teacher models with respect to student loss on quiz dataset on step 6.

4.2 Experiment Results

Table 1 shows experimental results for the models reported in terms of accuracy, number of parameters, and compression times. For the third and fourth block, this convection was followed for naming the student model: *teacher_name student_name*. For example, BERT DistilBERT in the third block denotes that DistilBERT was the student model taught by BERT as a teacher using normal knowledge distillation. For the model with the additional teacher model, this naming convention was followed for the student model: *teacher_name1* + *teacher_name2 student_name*. For the compression times, it was estimated by how many times each student model was smaller than the average number of teacher model's parameters.

From the first block, two versions of the BERT model are reported. 'BERT' model was used as a single teacher training while the 'BERT2' model was used only as an additional teacher model to the 'BERT' model. While 'BERT' and 'BERT2' use the same initialization, they were trained on different train datasets of 30,000 samples. The training dataset used by the 'BERT' model is used for all student models.

From the results, the student model as DistilBERT (in 2nd row of last block) trained by two BERT models ('BERT' and 'BERT2') on

MetaDistil framework gave the **best result** among the student models. It gave accuracy very close to BERT models while beating RoBERTa. This highlights the benefit of having an extra teacher model providing knowledge from its perspective.

It is noted that DistilBERT and TinyBERT trained under teacher model's supervision on distillation algorithms gave better accuracies than their fine-tuned counterparts. This further highlights the efficacy of having high performing teacher models bolstering the performance of student models.

However, it is the exception in having RoBERTa as the teacher model. It worsened the performance of student models compared to its fine-tuned model counterparts. This demonstrates the importance of selecting the right teacher model for the knowledge distillation to process effectively.

Another case where a student model trained via distillation process performed worse than its fine-tuned counterpart is the DistilBERT model (last row of 4th row) where it was fine-tuned for 1 epoch. This indicates fine-tuning as a warm up before distillation process is not effective for improving the student model's performance.

Finally, adding noise seems to drop the DistilBERT's performance compared to all student models. This can be attributed due to the nature of sentiment analysis tasks. Since certain words are vital for determining the sentiment, masking a word in a sentence could potentially cause that important information to be lost.

5. Discussion

In this paper, adding noise, additional teacher model, and fine-tuning for 1 epoch were explored for potential improvement on MetaDistil framework. Having an extra teacher proved to be an effective improvement with the right teacher models.

Unfortunately, the MetaDistil framework, considering only the student model trained by a single teacher model, did not do better than normal knowledge distillation. For 'BERT DistilBERT' model, both algorithms gave same accuracies while MetalDistil performed worse for 'BERT TinyBERT' model. It excelled for the 'RoBERTa DistilBERT' model where it did 7.8% accuracy better.

However, because MetaDistil excelled in one model only, this could undermine the credibility of having the need of MetaDistil for the model where an additional teacher model was incorporated where normal knowledge distillation could suffice with lesser training time and lesser memory consumption of GPU.

6. Conclusion

In this study, we implemented the MetaDistil framework and explored potential improvements to enhance its performance. We investigated the effects of noise injection, an additional teacher model, and fine-tuning for one epoch. We evaluated the performance of our approach against the normal knowledge distillation process and models that were directly fine-tuned, as well as the MetaDistil framework without considering these factors. Overall, our results demonstrated the potential of incorporating an extra teacher model to generate better student models; however, this improvement came at the cost of increased training time and memory usage. In conclusion, our study highlights the importance of considering noise injection, additional teacher models, and fine-tuning in the MetaDistil framework. These factors can lead to improved performance, although trade-offs in terms of training time and memory must be carefully considered.

7. Future Work

In future experimentation, adding teacher models for training a student model and many students learning together will be explored for teacher-student architecture. Also, more configurations of noise will be explored in an effort to make student models more robust and versatile. Finally, further consideration will be given to pass more teacher knowledge to the student model by incorporating knowledge from the teacher model's intermediate layers.

References

Gutierrez, R. L., & Leonetti, M. (2021). Information-theoretic Task Selection for Meta-Reinforcement Learning [Preprint]. *arXiv* preprint arXiv:2011.01054v2 [cs.LG]. https://doi.org/10.48550/arXiv.2011.01054

Romero, A., Ballas, N., Ebrahimi Kahou, S., Chassang, A., Gatta, C., & Bengio, Y. (2015). FitNets: Hints for Thin Deep Nets. *arXiv* preprint arXiv:1412.6550 [cs.LG]. https://doi.org/10.48550/arXiv.1412.6550

Jabri, A., Hsu, K., Gupta, A., Eysenbach, B., Levine, S., & Finn, C. (2019). Unsupervised Curricula for Visual Meta-Reinforcement Learning. In *Advances in Neural Information Processing Systems 32* (NeurIPS 2019).

Antoniou, A., Edwards, H., & Storkey, A. (2019). How to train your MAML. In *International Conference on Learning Representations (ICLR)*. arXiv preprint arXiv:1810.09502. https://doi.org/10.48550/arXiv.1810.09502

Bansal, T., Jha, R., Munkhdalai, T., & McCallum, A. (2020). Self-Supervised Meta-Learning for Few-Shot Natural Language Classification Tasks. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://doi.org/10.18653/v1/2020.emnlp-main.38

Peng, B., Jin, X., Li, D., Zhou, S., Wu, Y., Liu, J., Zhang, Z., & Liu, Y. (2019). Correlation Congruence for Knowledge Distillation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV).* https://doi.org/10.1109/iccv.2019.00511

Buciluǎ, C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 535-541. https://doi.org/10.1145/1150402.1150464

Chen, Y., Zhong, R., Zha, S., Karypis, G., & He, H. (2022). Meta-learning via Language Model In-context Tuning. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers). https://doi.org/10.18653/v1/2022.acl-long.53

Liu, C., Wang, Z., Sahoo, D., Fang, Y., Zhang, K., & Hoi, S. C. H. (2020). Adaptive Task Sampling for Meta-learning. *Computer Vision – ECCV 2020*, 752-769. https://doi.org/10.1007/978-3-030-58523-5_44

Finn, C., Rajeswaran, A., Kakade, S., & Levine, S. (2019). Online Meta-Learning. *arXiv* preprint arXiv:1902.08438 [cs.LG]. https://doi.org/10.48550/arXiv.1902.08438

Tung, F., & Mori, G. (2019). Similarity-Preserving Knowledge Distillation [Preprint]. arXiv:1907.09682v2 [cs.CV]. https://doi.org/10.48550/arXiv.1907.09682

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv* preprint arXiv:1503.02531 [stat.ML]. https://doi.org/10.48550/arXiv.1503.02531

Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge Distillation: A Survey. International Journal of Computer Vision. *Advance online publication.* https://doi.org/10.1007/s11263-021-01453-z

Pan, H., Wang, C., Qiu, M., Zhang, Y., Li, Y., & Huang, J. (2021). Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 3026–3036. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.236

Chen, P., Liu, S., Zhao, H., & Jia, J. (2021). Distilling Knowledge via Knowledge Review. *arXiv* preprint arXiv:2104.09044 [cs.CV]. https://doi.org/10.48550/arXiv.2104.09044

Rajasegaran, J., Khan, S., Hayat, M., Khan, F. S., & Shah, M. (2020). Self-supervised knowledge distillation for few-shot learning. *arXiv* preprint arXiv:2006.09785. https://doi.org/10.48550/arXiv.2006.09785

Kaddour, J., Sæmundsson, S., & Deisenroth, M. P. (2020). Probabilistic Active Meta-Learning. Retrieved from *arXiv* preprint arXiv:2007.08949. (Version 2) doi: https://doi.org/10.48550/arXiv.2007.08949

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for natural language understanding, *arXiv* preprint arXiv:1909.10351v5. https://doi.org/10.48550/arXiv.1909.10351

Lee, H.-y., Li, S.-W., & Vu, N. T. (2022). Meta Learning for Natural Language Processing: A Survey (Version 2). *arXiv* preprint arXiv:2205.01500v2 [cs.CL]. https://doi.org/10.48550/arXiv.2205.01500

Huang, Z., & Wang, N. (2017). Like What You Like: Knowledge Distill via Neuron Selectivity Transfer . *arXiv*. Retrieved from https://arxiv.org/abs/1707.01219

Liu, Y., Zhang, W., & Wang, J. (2021). Adaptive Multi-Teacher Multi-level Knowledge Distillation . *arXiv* preprint arXiv:2103.04062.

Liu, Y., Shen, S., & Lapata, M. (2021). Noisy Self-Knowledge Distillation for Text Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 692-703.

Ko, H.-J., Lee, J., Kim, J., Lee, J., & Shim, H. (2020). Diversity regularized autoencoders for text generation. *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 883-891. https://doi.org/10.1145/3341105.3373998

Meng, Z., Li, J., Zhao, Y., & Gong, Y. (2019). Conditional Teacher-Student Learning. *arXiv* preprint arXiv:1904.12399 [cs.LG]. https://doi.org/10.48550/arXiv.1904.12399

Pan, H., Wang, C., Qiu, M., Zhang, Y., Li, Y., & Huang, J. (2022). Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains. *arXiv* preprint arXiv:2012.01266v2 [cs.CL]. https://doi.org/10.48550/arXiv.2012.01266

Zagoruyko, S., & Komodakis, N. (2017). Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer (Version 3). *arXiv* preprint arXiv:1612.03928v3 [cs.CV]. https://arxiv.org/abs/1612.03928v3

Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., & Ghasemzadeh, H. (2020). Improved Knowledge Distillation via Teacher Assistant. *Proceedings of the AAAI Conference* *on Artificial Intelligence*, 34(04), 6263-6270. https://doi.org/10.1609/aaai.v34i04.5963.

Sun, S., Cheng, Y., Gan, Z., & Liu, J. (2019). Patient Knowledge Distillation for BERT Model Compression. *arXiv* preprint arXiv:1908.09355 [cs.CL]. https://doi.org/10.48550/arXiv.1908.09355

Tian, Y., & Zhao, X. (2022). Meta-learning approaches for learning-to-learn in deep learning: A survey. *Neurocomputing*, 494(19). https://doi.org/10.1016/j.neucom.2022.04.078

Zhou,Wangchunshu, Canwen Xu, & Julian McAuley. (2022). BERT Learns to Teach: Knowledge Distillation with Meta Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 7037-7049. https://doi.org/10.18653/v1/2022.acl-long.485

Park, W., Kim, D., Lu, Y., & Cho, M. (2019). Relational Knowledge Distillation. *arXiv* preprint arXiv:1904.05068 [cs.CV]. Retrieved from https://arxiv.org/abs/1904.05068

Li, X., Yu, L., Jin, Y., Fu, C.-W., Xing, L., & Heng, P.-A. (2020). Difficulty-aware Meta-learning for Rare Disease Diagnosis. *arXiv* preprint arXiv:1907.00354v2 [cs.CV]. https://doi.org/10.48550/arXiv.1907.00354

Liu, Y., Shen, S., & Lapata, M. (2021). Noisy Self-Knowledge Distillation for Text Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 692-703. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.56

You, S., Xu, C., Xu, C., & Tao, D. (2017). Learning from Multiple Teacher Networks. In *KDD '17: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1285-1294. https://doi.org/10.1145/3097983.3098135

Zhang, Y., Xiang, T., Hospedales, T. M., & Lu, H. (2017). Deep Mutual Learning. *arXiv* preprint arXiv:1706.00384 [cs.CV]. https://doi.org/10.48550/arXiv.1706.00384

Zhao, J., Luo, W., Chen, B., & Gilman, A. (2021). Mutual-Learning Improves End-to-End Speech Translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3989-3994. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.3

Wu, C., Wu, F., & Huang, Y. (2021). One teacher is enough? pre-trained language model distillation from multiple teachers. *arXiv preprint arXiv:2106.01023*.

Ilichev, A., Sorokin, N., Piontkovskaya, I., & Malykh, V. (2021). Multiple Teacher Distillation for Robust and Greener Models. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, 601-610.