Unified Distillation: Supervising the Student to Make Fewer Mistakes

Yuan Chen and Pan Rong
School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China.
Email: yuanch23@mail2.sysu.edu.cn, panr@sysu.edu.cn

Abstract. Knowledge distillation has become a popular task in modern deep learning applications by performing knowledge transfer from a cumbersome neural network commonly called “teacher model” to a much smaller network called “student model”. In traditional knowledge distilling process, there are usually two objectives for training the student model, namely hard target and soft target. However, sometimes it’s hard to find a trade-off between them. We unify the two objectives into one making it easier to perform knowledge distillation, and propose a novel distilling method called “Unified Distillation” to supervise the student to make fewer mistakes. The model can correct the wrong predictions according to the hard target, and maintain the advantage of knowledge distillation. Although our method can be used in almost all fields suitable for knowledge distillation, we choose neural machine translation as a study object for its complexity. We conducted experiments on three neural machine translation tasks, using a finetuned language model BERT as the teacher, and a Transformer base model as the student. The experimental results indicate that our method is better than traditional knowledge distillation method.

Keywords. Unified Distillation, knowledge distillation, neural machine translation.

1. Introduction

Nowadays, deep learning has become ubiquitous in various research fields such as computer vision (CV) and nature language processing (NLP). However, researcher’s pursuit of high-performance computing makes deep neural networks more and more complex so that these networks need more storage and computing resources. As a result, the extremely large model size prevents these neural networks from being deployed and operated in devices with poor computation power such as mobile phones or other handheld devices. So knowledge distillation (KD) came to the rescue of this problem and became one of the most widely-used model compression techniques. The main idea of KD is transfer learning, training a simplified model commonly referred to as the student model to mimic the behavior of the complex pre-trained model commonly called the teacher model.

Cristian Bucilua called the dataset for training the student model a “transfer set” instead of a general training set [1]. Although the transfer sets do not require having ground truth labels, researchers generally
have the transfer sets with ground truth labels available and these labels offer much help for model performance improvement [2][3][4][5][6][7][8]. In this situation, there are two objectives that the student model needs to aim at, one is $L_{soft}$, which measures the difference between the output probability distributions of the teacher and that of the student, and the output probability distribution of the teacher are commonly called “soft target”; the other is $L_{hard}$, which measures the difference between one-hot encoding label of ground-truth called “hard target” and the output probability distributions of the student. And the both objectives are combined as Equation (1).

$$L = \alpha L_{soft} + (1 - \alpha)L_{hard}.$$  

The ratio of two objectives is tuned with a hyper-parameter $\alpha$. Generally, $\alpha$ is set to 0.5 by default, but the best practice is to set $\alpha$ to a larger value to emphasize $L_{soft}$ [3].

Equation (1) is widely used in KD applications [2][4][3][5][7][8]. However, it appears to be inelegant and casual to simply add $L_{soft}$ and $L_{hard}$ in a fixed proportion. If the teacher’s prediction is consistent with the ground-truth label, it seems unnecessary and redundant to use $L_{hard}$. Otherwise, the student may get confused on which one of the two objectives to aim at. Just as an old proverb goes, “Person who chases two rabbits catches neither.”

We propose a novel distilling method called “Unified Distillation” in order to disambiguate the two training objectives. Our method is shown in Figure 1. We first added the hard and the soft target distributions together as one combined target. Notice that the combined target is not a probability distribution before normalized, because the sum of all probability values doesn’t equal to 1. The max probability value of the combined target can be very large or not very large depending on whether the teacher's prediction is in consistent with the ground-truth label or not, so we need to rectify it by rescaling the max probability value to a fixed value, which is given by a hyper-parameter $P_{max}$. Rectifying the combined target serves two purposes: (1) We can convert it into a normalized probability distribution; and (2) We can make the target distribution smoother. Thus, not only can we maintain the supervision of ground-truth label, but also highlight the information that the small probability values contain. Negative labels with small probability values contain plenty of information, which can be viewed as knowledge of the teacher. The corresponding probability values of some negative labels may be greater than that of other negative labels. For example, in an image classification task, the similarity between a truck and a garbage truck is much greater than that between a truck and a carrot [3].

![Figure 1](image1.png)

**Figure 1.** Illustration of our method. We first combined the soft and hard targets together directly, and then rectified the max value of probability to $P_{max}$.

Figure 2 shows the comparison between our method and general knowledge distillation method with $\alpha$ set to 0.2, 0.5 and 0.8, respectively. The top half of the figure represents the situation when the teacher’s prediction is consistent with the hard target, and the bottom half of the figure shows the opposite situation.
In general knowledge distillation, we optimize student’s output distribution to minimize $L$ in Equation (1). When $\alpha$ is set to 0.2, the influence of hard target is emphasized. Although the mistakes made by the teacher can be corrected, when the teacher’s prediction is consistent with the hard target, the valuable small probability values are diminished, and the student is hard to obtain knowledge in this way. When $\alpha$ is set to 0.5, the influences of both soft and hard target are equal. The mistakes made by the teacher can be corrected, but the probability value of the mistaken prediction label only slightly falls short of that of the correct label, and the student may get confused on which labels to follow. If the teacher’s prediction is consistent with the hard target, the valuable small probability values are also diminished but not as severe as the case when $\alpha$ is 0.2, which may hamper the student from learning knowledge from teacher. When $\alpha$ set to 0.8, the soft target is highlighted, but the mistakes cannot be corrected. In knowledge distillation, the teacher is much more complex and powerful than the student, so the proportion of correct predictions is always much higher than the incorrect ones. Consequently, the best practice is to set $\alpha$ to a higher value to prefer the major correct proportion. Obviously, one fixed value of $\alpha$ cannot coordinate the contradiction among the two targets and the correct or incorrect prediction. Our Unified Distillation fabricates a virtual “teacher” which always makes correct predictions by combining both soft and hard targets and forming a unified target. The Unified target is always correct and does not diminish the valuable small probability values regardless of whether the teacher’s prediction is correct or not.

![Figure 2. Comparison between our method (middle) and general KD method (right) in both cases.](image)

Although our “Unified Distilling” is a general method and can be used in almost all fields suitable for knowledge distillation, we choose neural machine translation (NMT) as the object of study for its complexity. And we believe that it works well in other fields if it works in NMT tasks.

We used a finetuned language model BERT as the teacher, which can generate sequences of word probability values as soft targets for the training samples in KD process, and a Transformer base model as the student, which can learn from the teacher’s outputs. Experimental results indicate that our method outperforms the general KD method, the Transformer baselines, and the proposed method that dynamically adjust distillation temperature [9].

2. Related works
Knowledge distillation was first proposed by Bucilua et al. in 2006 [1] and further made popular by Ba and Caruana in 2014 [10] and then Hinton et al. in 2015 [3]. Bucilua et al. firstly attempted to use the outputs of a network to train another neural network [1]. Ba and Caruana further used the logits in a cumbersome neural network as supervision to train a much smaller network so that the model size can be reduced [10]. Hinton et al. systematically summarized a widely used method of KD by dividing the logits by an extra hyper-parameter called “temperature” so that the output distributions can be smoothed [3]. After that, KD has become one of the important methods of model compression.

Many studies sprung up to enrich and improve the supervisory information of training the student models for distilling knowledge efficiently in both CV and NLP tasks [1][4][12]. Some studies showed that the teacher’s output in the intermediate hidden layers can also provide essential information [12]. Tiancheng Wen et al. proposed a knowledge adjustment method that generated better supervision for student model, increasing accuracy on image classification datasets [9]. In NLP field, some works utilized the knowledge from the teacher at different levels jointly, such as embedding, representations output by hidden layers, the output of self-attention modules as well as the final predictions of the teacher model [13][14].

NMT task is to predict the text sequence, in which each word is based on previous words, and finally generate the whole sequence. So NMT models are always so large that KD receives much attention. Yoon Kim et al. conducted KD not only at word level, but also used beam search at sequence level, compressing a LSTM based NMT model [8]. Yen-Chun Chen et al. proposed a new Conditional Masked Language Modeling (C-MLM) method, which enabled fine-tuning pre-trained BERT on a target dataset and leveraging it into text generate tasks such as NMT [7].

3. Methodology
We first detail our Unified Distillation method in 3.1, and then present the application of Unified Distillation for NMT Tasks in 3.2.

3.1 Unified Distillation
There are two steps in Unified Distillation method. We first compute the combined target by directly adding probability distribution namely soft target from the teacher and one-hot label together as Equation (2).

\[ z(x) = p^t(x) + l(x), \]  

(2)

where \( z(x) \) and \( p^t(x) \) denote the combined target and the output probability distribution of the teacher on sample \( x \) in a batch, respectively. \( l(x) \) is the one-hot label of sample \( x \), and the \( i \)-th term in \( l(x) \) is defined as Equation (3).

\[ l_i(x) = \begin{cases} 1, & \text{if } i = c(x) \\ 0, & \text{if } i \neq c(x) \end{cases}, \]  

(3)

where \( c(x) \) denotes the correct class of sample \( x \).

The second step is to rectify the combined target, as is shown in Figure 1. Obviously, the maximum value of \( z(x) \) corresponds to the correct class. We rectified the combined target to make Equation (4) hold by tuning \( \tau_x \),

\[ \frac{\exp(\max(z(x))/\tau_x)}{\sum \exp(z(x)/\tau_x)} = P_{\text{max}}, \]  

(4)

where \( \tau_x \) denotes the temperature parameter for sample \( x \), and hyper-parameter \( P_{\text{max}} \) is set in the range of 0.5 to 1.
Since Equation (4) has no closed-form solution, $\tau_x$ cannot be computed directly. We built a small machine learning model with the objective of minimizing loss function $L_\tau$ in order to obtain approximate solution of $\tau_x$ to satisfy Equation (5),

$$L_\tau = \frac{1}{N} \times \sum_x \text{MSE}\left(\frac{\exp(\max(x(x))/\tau_x)}{\sum \exp(x(x)/\tau_x)}, P_{\text{max}}\right),$$

where $N$ is batch size of samples, and MSE stands for the mean squared error loss function.

Algorithm 1 gives the process of solving $\tau_x$, which consists $N$ elements of $\tau_x$ in a batch. This algorithm doesn’t significantly increase the time of training the student model for we set $\text{max}_\text{iter}$ to a small value like 10.

### ALGORITHM 1: Solve $\tau_x$ for Rectifying the Combined Target

The algorithm receives as input a batch of samples $X$, the output probability distribution of the teacher $p^t(X)$, the one-hot labels of samples in a batch $l(X)$, the maximum number of iteration $\text{max}_\text{iter}$, the hyper-parameter $P_{\text{max}}$ and the learning rate $\eta$. The algorithm applies gradient descent method iteratively to solving $\tau_x$ and returns the solved values.

**procedure** $\text{solve}_\tau(x, p^t(X), l(X), \text{max}_\text{iter}, P_{\text{max}}, \eta)$

1. $N \leftarrow \text{length}(X)$
2. $\tau_x \leftarrow$ Initialized temperature parameters for a batch
3. for 1, 2, …, $\text{max}_\text{iter}$ do
   1. for each $x$ in $X$ do
      1. calculate $z(x)$ using Equation (2)
      2. calculate $L_\tau$ using Equation (5)
      3. $\tau_x \leftarrow \tau_x - \eta \frac{\partial L_\tau}{\partial \tau_x}$
4. return $\tau_x$

The solved sample-wise temperature $\tau_x$ can be applied to Equation (6) to compute the unified target $p^{\text{unified}}(x)$ for sample $x$, and the student model can be trained using loss function given by Equation (7),

$$p^{\text{unified}}(x) = \frac{\exp(z(x)/\tau_x)}{\sum \exp(z(x)/\tau_x)},$$

$$L^{\text{unified}} = \frac{1}{N} \times \sum_x \text{KL}(p^s(x), p^{\text{unified}}(x)),$$

where $p^s(x)$ denotes the output probability distribution of the student, and $\text{KL}$ stands for Kullback-Leibler divergence.

### 3.2 Unified Distillation for NMT tasks
In order to verify the effectiveness of our proposed method, we performed Unified Distillation on NMT tasks. We followed the basic teacher-student framework in [7], applying a finetuned language model BERT as the teacher and a Transformer base model as the student, except that the general KD method was changed into Unified Distillation method illustrated in Section 3.1, as shown in Figure 3.

We chose this teacher-student framework for two reasons: (1) It achieved competitive results on several NMT tasks, especially on IWSLT German-English dataset; and (2) The general KD method of combining both soft and hard targets together was applied in this framework, so it's easy to compare our method with the general KD method to verify that our method is superior.

Generally, BERT is trained with a generative objective via MLM during the pre-training stage, and this training objective forces the student model to learn essential, bidirectional, contextual knowledge. Because BERT is not auto-regressive enabled, it cannot be directly applied to auto-regressive NMT in practice. Y. C. Chen proposed a novel method called C-MLM to fine-tune the pre-trained BERT model, which is similar to MLM but requires additional conditional inputs [7]. We applied C-MLM to fine-tuning the pre-trained BERT-base-multilingual-cased model on the target datasets. After the BERT model was finetuned, we set it as the teacher that can generate probability distributions of tokens (soft targets) for the training samples, and set the Transformer base model as the student. Because both the left and the right contexts of each token can be fully exploited in the finetuned BERT model, the single-directional limitation of general Seq2Seq training framework can be compensated, and the student can effectively learn more information from the teacher’s outputs. More details can be found in [10] and [7].

To compute the soft targets of the finetuned BERT teacher, we applied the circular mask strategy proposed in [7], and obtained probability distributions predicted by the BERT teacher for each output target tokens in masked position as soft targets. The whole process of computing soft targets by circular mask is shown in Figure 3.
In an NMT task, a pair of source and target sentences is processed as a sample, and each sentence or sequence is consist of tokens. We perform Unified Distillation for each token at a time in one target sequence. Take the third token as an example, as is shown in Figure 4, we combined the probability distribution of masked token $t_3$ computed by the teacher and the known one-hot label to obtain the combined target. Then the temperature value for the token $t_3$ can be solved using Algorithm 1 so that the combined target can be rectified. Thus, the unified target is obtained, which can supervise the training of the student model.

![Figure 4](image)

**Figure 4.** The process of computing the soft target for each token in target sequence.

The subscript character $n$ denotes the length of the target sentence. A circular mask strategy is applied, and repeating in a training pair for 7 times to approximate the 15% masking rate. The masked pair is input to the finetuned BERT in every iteration, and the probability distribution of each masked token is computed and outputted.

At inference stage, the BERT teacher is not used so that the decoding speed is the same as the student Transformer model alone.

### 4. Experiments and Results

We conducted experiments on three NMT tasks, using datasets and basic experimental settings described in 4.1. Detailed experiments and results are given, and BLEU scores are presented for evaluation of NMT.

#### 4.1 Datasets and Basic Settings

We considered two relatively small-scale datasets, IWSLT14 German-English (De-En, 160k training samples) and IWSLT15 English-Vietnamese (En-Vi, 113k training samples) [15] and one relatively large-scale dataset, UM-Corpus (En-Zh, 2.2M training samples) [14]. For IWSLT14 De-En, we followed the pre-processing steps presented in [6]. For IWSLT15 En-Vi, we used tst2012 as dev set and tst2013 as test set. For UM-Corpus En-Zh, we used UM-Testing dataset that contains 5,000 samples as the test set followed the settings of [14].

| Dataset            | Total Steps | warmup_steps | factor | Batch size |
|--------------------|-------------|--------------|--------|------------|
| IWSLT14 De-En      | 100k        | 8k           | 1      | 6k         |
| IWSLT15 En-Vi      | 30k         | 8k           | 1      | 6k         |
| UM-Corpus En-Zh    | 500k        | 40k          | 1.5    | 2k         |

We adapted a pre-trained BERT-base-multilingual-cased model by fine-tuning it with C-MLM as the teacher, and a Transformer base of 6 encoder/decoder blocks and 8 attention heads attention blocks as the student [3], where the size of hidden feed-forward layer is 2048 and the size of hidden size is 512. BERT
tokenizer was used for word segmentation, and the training settings listed in Table 1 were applied to our experiments for three datasets. Two GeForce GTX 1080ti GPUs were used in our experiments.

For all experiments, the learning rate schedule is set to Equation (8) according to [3],

$$lr = \text{factor} \times d_{\text{model}}^{-0.5} \times \min(\text{step}^{-0.5}, \text{step} \times \text{warmup steps}^{-1.5})$$

(8)

where $d_{\text{model}}$ is the hidden size, and $\text{factor}$ is a hyper-parameter.

Before training, pairs of source and target sequences in training set are preprocessed and batched sequences of similar length, and batch size is counted by the number of tokens instead of the number of pairs. In KD process, the BERT’s prediction logits of the training data were computed, and we used top-K distillation [20] where K is set to 8 through all the experiments. The temperature value $\tau_x$ for each token is initialized to 1.

4.2 Experimental Results

We evaluated Unified Distillation on IWSLT14 German-English translation, IWSLT15 English-Vietnamese translation, and UM-Corpus English-Chinese translation datasets. The evaluation results are presented in Table 2, Table 3, and Table 4. The value of hyper-parameter $P_{\text{max}}$ should be tuned according to the dataset.

Table 2 indicates that on IWSLT14 German-English translation dataset with $P_{\text{max}}$ set to 0.75, our United Distillation method outperforms the results of Transformer baseline and Transformer trained with general KD method by 4.9% and 0.2% on development set, respectively; 5.5% and 0.8% on test set, respectively.

Table 3 demonstrates that on IWSLT15 English-Vietnamese translation dataset with $P_{\text{max}}$ set to 0.90, our United Distillation method clearly outperforms the results of RNN baseline and RNN trained using the general KD method presented in [7] by 19.0% and 10.9% on development set, respectively; 17.3% and 13.9% on test set, respectively. However, our United Distillation method is still at par with the results presented in the same literature. Because of the limited computing power on our available computer, we set the batch size much smaller than that of original experiment, perhaps that’s the reason why our method does not achieve prominent results on this dataset.

Table 4 shows that on UM-Corpus English-Chinese translation dataset, our United Distillation method outperforms SMT baseline result presented in [14] by 0.78% with $P_{\text{max}}$ set to 0.80.

| Method                              | BLEU (Dev) | BLEU (Test) |
|-------------------------------------|------------|-------------|
| Unified Distillation ($P_{\text{max}}$=0.75) | 37.00      | 35.92       |
| Transformer only†                   | 35.27      | 34.04       |
| Transformer + BERT teacher†         | 36.93      | 35.63       |

| Method                              | BLEU (Dev) | BLEU (Test) |
|-------------------------------------|------------|-------------|
| Unified Distillation ($P_{\text{max}}$=0.90) | 27.88      | 31.44       |
| RNN only†                           | 23.37      | 26.80       |
| RNN + BERT teacher†                 | 25.14      | 27.59       |
| Transformer only†                   | 27.03      | 30.76       |
| Transformer + BERT teacher†         | 27.85      | 31.51       |
Table 4: BLEU scores for UM-Corpus English-Chinese translation. (‡) from [14].

| Method                                      | BLEU (Test) |
|---------------------------------------------|-------------|
| Unified Distillation (Our method with $P_{\text{max}}=0.80$) | 31.81       |
| SMT‡                                        | 31.55       |

4.3 Effects of $P_{\text{max}}$

We conducted three NMT tasks experiments with different values of $P_{\text{max}}$, and the results are listed in Table 5.

Table 5 Experimental results on three NMT datasets using different values of $P_{\text{max}}$.

| $P_{\text{max}}$ | IWSLT14 De-En | IWSLT15 En-Vi | UM-Corpus En-Zh |
|------------------|----------------|---------------|-----------------|
|                  | BLEU (Dev)    | BLEU (Test)   | BLEU (Dev)     | BLEU (Test)    |
| 0.95             | 35.12         | 33.67         | 27.41          | 29.72          | 31.60          |
| 0.90             | 35.35         | 33.76         | 27.88          | 31.44          | 31.56          |
| 0.85             | 35.88         | 34.23         | 27.67          | 30.95          | 31.74          |
| 0.80             | 36.73         | 35.56         | 27.53          | 31.09          | 31.81          |
| 0.75             | 37.00         | 35.92         | 27.51          | 31.22          | 31.65          |
| 0.70             | 36.81         | 35.64         | 27.59          | 31.49          | 31.46          |
| 0.65             | 36.78         | 35.63         | 27.42          | 31.43          | 31.02          |
| 0.60             | 36.50         | 35.39         | 27.33          | 30.69          | 30.90          |

For IWSLT14 De-En, when $P_{\text{max}}$ is set to 0.75, student Transformer model achieves the highest BLEU scores on both dev set and test set. For IWSLT15 En-Vi, the best BLEU scores are obtained on dev and test set when $P_{\text{max}}$ is set to 0.90 and 0.70, respectively. For UM-Corpus En-Zh, when $P_{\text{max}}$ is set to 0.80, student model achieves the highest BLEU score on test set. The results confirms that the value of $P_{\text{max}}$ needs to be tuned according to different datasets.

Table 6. Experimental results of our method and other different temperature adjustment methods, •† is from [9].

| Method                | IWSLT14 De-En | IWSLT15 En-Vi | UM-Corpus En-Zh |
|-----------------------|---------------|---------------|-----------------|
|                       | BLEU (Dev)    | BLEU (Test)   | BLEU (Dev)     | BLEU (Test)    |
| United Distillation   | 37.00         | 35.92         | 27.88          | 31.44          | 31.81          |
| Fixed value ($\tau=1$)| 35.35         | 34.30         | 26.67          | 29.95          | 30.49          |
| CWSM†                 | 35.98         | 34.82         | 27.18          | 30.04          | 31.33          |
| FLSW†                 | 35.85         | 34.65         | 27.05          | 29.99          | 31.10          |

4.4 More Experiments

More experiments were conducted to compare the Unified Distillation method with the general KD method using different temperature adjustment strategies. The results are listed in Table 6.

The results show that our Unified Distillation method outperforms the method using the fixed value 1 (the value for initializing all temperature values in our method) for each token as well as other dynamic temperature adjustment methods proposed in [9].
5. Conclusion and Discussion

We proposed a novel method called “Unified Distillation” to improve the general KD method by combining two objectives together and then rectifying it by rescaling the max probability to a fixed value. In this way, the information of ground-truth label and the “dark knowledge” from the teacher model can be unified into one target, avoiding the dilemma of chasing two targets at the same time. We conducted NMT experiments on three datasets to validate our method. A finetuned language model BERT is adopted as a teacher, and a Transformer base model is adopted as a student. Experimental results indicate that our method provides a better approach for knowledge distillation applications.

For future work, we will try some of the most representative algorithms to solve the problem, such as krill herd [22], monarch butterfly optimization (MBO) [21], earthworm optimization algorithm (EWA) [23], elephant herding optimization (EHO) [24] and moth search (MS) algorithm [25].

REFERENCES

[1] Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model Compression. In Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[2] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for Thin Deep Nets. In International Conference on Learning Representations (ICLR). arXiv: 1412.6550v4.

[3] Geoffrey Hinton, Oriol Vinyals, Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. In Computer Science, 2015, 14(7), pp. 38-39.

[4] Junho Yim, Donggyu Joo, Jihoon Bae, Junno Kim. 2017. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[5] Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, Jimmy Lin. 2019. Distilling Task-Specific Knowledge from BERT into Simple Neural Networks. arXiv: 1903.12136.

[6] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. arXiv: 2002.10957.

[7] Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, Jingjing Liu. 2019. Distilling Knowledge Learned in BERT for Text Generation. arXiv: 1911.03829v2.

[8] Yoon Kim and Alexander M. Rush. 2016. Sequence-level Knowledge Distillation. arXiv:1606.07947v4.

[9] Tiancheng Wen, Shenqi Lai, Xueming Qian. 2019. Preparing Lessons: Improve Knowledge Distillation with Better Supervision. arXiv:1911.07471.

[10] Lei Jimmy Ba, Rich Caruana. 2014. Do Deep Nets Really Need to Be Deep? In Advances in Neural Information, Processing Systems 27, pages 2654–2662. Curran Associates, Inc. arXiv: 1312.6184v1.

[11] Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan Yuille. 2018. Knowledge Distillation in Generations: More Tolerant Teachers Educate Better Students. arXiv: 1805.05551.

[12] Sergey Zagoruyko, Nikos Komodakis. 2017. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In ICLR 2017.

[13] Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf. 2019. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. arXiv: 1910.01108.

[14] Xiaojing Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang and Qun Liu. 2019. TinyBERT: Distilling BERT for Natural Language Understanding. arXiv: 1909.10351.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.
[16] Minh-Thang Luong, Christopher D. Manning. 2015. Stanford Neural Machine Translation Systems for Spoken Language Domains. In International Workshop on Spoken Language Translation (IWSLT).
[17] Liang Tian, Derek F. Wong, Lidia S. Chao, Paulo Quaresma, Francisco Oliveira, Yi Lu, Shuo Li, Yiming Wang, Longyue Wang. 2014. UM-Corpus: A Large English-Chinese Parallel Corpus for Statistical Machine Translation”. In Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC’14), Reykjavik, Iceland.
[18] Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, Michael Auli. 2019. Pay Less Attention with Lightweight and Dynamic Convolutions. arXiv: 1901.10430.
[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. 2017. Attention Is All You Need. In 31st Conference on Neural Information Processing Systems (NIPS 2017). arXiv: 1706.03762.
[20] Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao and Tie-Yan Liu. 2019. Multilingual Neural Machine Translation with Knowledge Distillation. In ICLR, 2019.
[21] Gai-Ge Wang, Suash Deb and Zhihua Cui. 2019. Monarch Butterfly Optimization. Neural Computing and Applications 31, 1995–2014.
[22] A, Amir Hossein Gandomi, and A. H. A. B . 2012. Krill Herd: A New Bio-Inspired Optimization Algorithm. Communications in Nonlinear Science and Numerical Simulation 17. 12(2012):4831-4845.
[23] Khan, Sajawal Ur Rehman, et al. 2017. Genetic Algorithm and Earthworm Optimization Algorithm for Energy Management in Smart Grid. 12th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2017), Spain Springer, Cham.
[24] Gai-Ge Wang, Suash Deb and Leandro Coelho. 2015. Elephant Herding Optimization. 2015 3rd International Symposium on Computational and Business Intelligence (ISCI1), Bali, Indonesia, 2015, pp. 1-5, doi: 10.1109/ISCI1.2015.8.
[25] Pyk P, Sergi Bermúdez i Badia, Bernardet U, et al. 2006. An artificial moth: Chemical source localization using a robot based neuronal model of moth optomotor anemotactic search[J]. Autonomous Robots, 2006, 20(3):197–213.