Sparse Teachers Can Be Dense with Knowledge

Conference or Workshop Item

How to cite:

For guidance on citations see FAQs.

© 2022 Association for Computational Linguistics

https://creativecommons.org/licenses/by/4.0/

Version: Version of Record

Link(s) to article on publisher’s website:
https://aclanthology.org/2022.emnlp-main.258

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online’s data policy on reuse of materials please consult the policies page.
Sparse Teachers Can Be Dense with Knowledge

Yi Yang*, Chen Zhang*, Dawei Song*
Beijing Institute of Technology
{yang.yi,czhang,dwsong}@bit.edu.cn

Abstract

Recent advances in distilling pretrained language models have discovered that, besides the expressiveness of knowledge, the student-friendliness should be taken into consideration to realize a truly knowledgeable teacher. Based on a pilot study, we find that over-parameterized teachers can produce expressive yet student-unfriendly knowledge and are thus limited in overall knowledgeableness. To remove the parameters that result in student-unfriendliness, we propose a sparse teacher trick under the guidance of an overall knowledgeable score for each teacher parameter. The knowledgeable score is essentially an interpolation of the expressiveness and student-friendliness scores. The aim is to ensure that the expressive parameters are retained while the student-unfriendly ones are removed. Extensive experiments on the GLUE benchmark show that the proposed sparse teachers can be dense with knowledge and lead to students with compelling performance in comparison with a series of competitive baselines.¹

1 Introduction

Pretrained language models (LMs) built upon transformers (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020) have achieved great successes. However, the appealing performance is usually accompanied with expensive computational costs and memory footprints, which can be alleviated by model compression (Ganesh et al., 2021). Knowledge distillation (Hinton et al., 2015), as a dominant method in model compression, concentrates on transferring knowledge from a teacher of large scale to a student of smaller scale.

Conventional studies (Sun et al., 2019; Jiao et al., 2020) mainly expect that the expressive knowledge would be well transferred, yet largely neglecting the existence of student-unfriendly knowledge. Recent attempts (Zhou et al., 2022; Zhao et al., 2022) are made to adapt the teacher to more student-friendly knowledge and have yielded performance gains. Based on these observations, we posit that over-parameterized LMs, on the one hand, can produce expressive knowledge due to over-parameterization, but on the other hand can also produce student-unfriendly knowledge due to over-confidence (Hinton et al., 2015; Pereyra et al., 2017). From a pilot study shown in Figure 1, we find that LMs of large scale tend to have a good performance and high confidence, and that both performance and confidence can be degraded through

Figure 1: Performance and confidence on RTE (Bentivogli et al., 2009) of BERT_{base} and BERT_{large} at small sparsity levels. Task metric and output distribution variance are used as the measures of performance and confidence, respectively. Distribution variance is comparatively equivalent to distribution negative entropy as employed in Pereyra et al. (2017). Proof of the equivalence can be found in Appendix A.
randomly sparsifying a small portion of parameters. This indicates that some parameters resulting in student-unfriendliness can be rather removed, to improve student-friendliness of the teacher without sacrificing too much its expressiveness.

Motivated by this finding, we propose a sparse teacher trick (in short, STARK) under the guidance of an overall knowledgeable score for each teacher parameter, which accords not only with the expressiveness but also the student-friendliness of the parameter by interpolation. The aim is to retain the expressive parameters while removing the student-unfriendly ones. Specifically, we introduce a three-stage procedure consisting of 1) trial distillation, 2) parameter sparsification, and 3) actual distillation. The trial distillation distills the dense teacher to the student so that a trial student is obtained. The parameter sparsification first estimates the expressiveness score and student-friendliness score of each teacher parameter via feedbacks respectively from the teacher itself and the trial student, and then sparsifies the teacher by removing the parameters associated with adequately low interpolated knowledgeable scores. The actual distillation distills the sparsified teacher to the student so that an actual student is obtained, where the student is initialized in the same manner as that used in trial distillation following the commonly-used rewinding technique (Frankle and Carbin, 2019).

We conduct an extensive set of experiments on the GLUE benchmark. Experimental results demonstrate that the sparse teachers can be dense with knowledge and lead to a remarkable performance of students compared with a series of competitive baselines.

2 Background

2.1 BERT Architecture

The BERT (Devlin et al., 2019) is composed of several stacked encoder layers of transformers (Vaswani et al., 2017). There are two blocks in every encoder layer: a multi-head self-attention block (MHA) and a feed-forward network block (FFN), with a residual connection and a normalization layer around each.

Given an \( l \)-length sequence of \( d \)-dimensional input vectors \( X \in \mathbb{R}^{l \times d} \), the output of the MHA block with \( A \) independent heads can be represented as:

\[
\text{MHA}(X) = \sum_{i=1}^{A} \text{Attn}(X, W_{Q}^{(i)}, W_{K}^{(i)}, W_{V}^{(i)})W_{O}^{(i)},
\]

where the \( i \)-th head is parameterized by \( W_{Q}^{(i)}, W_{K}^{(i)}, W_{V}^{(i)} \in \mathbb{R}^{d \times d} \), and \( W_{O}^{(i)} \in \mathbb{R}^{d \times d} \). On the other hand, the output of the FFN block is:

\[
\text{FFN}(X) = \text{GELU}(XW_{1})W_{2},
\]

where two fully-connected layers are parameterized by \( W_{1} \in \mathbb{R}^{d \times d} \) and \( W_{2} \in \mathbb{R}^{d \times d} \) respectively.

2.2 Knowledge Distillation

Knowledge distillation (Hinton et al., 2015) aims to transfer the knowledge from a large-scale teacher to a smaller-scale student, which is originally proposed to supervise the student with the teacher logits. With its prevalence, a tremendous amount of work has been investigated to transfer various knowledge from the teacher to the student (Romero et al., 2015; Zagoruyko and Komodakis, 2017; Sun et al., 2019; Jiao et al., 2020; Park et al., 2021b; Li et al., 2020; Wang et al., 2020). PKD (Sun et al., 2019) introduces a patient distillation scheme where the student learns multiple intermediate layer representations and logits from the teacher. Moreover, attention distributions (Sun et al., 2020; Jiao et al., 2020; Li et al., 2020; Wang et al., 2020) and even high-order relations (Park et al., 2021b) are considered to further boost the performance.

Since a large capacity gap between the teacher and the student can lead to an inferior distillation quality, TAKD (Mirzadeh et al., 2020) proposes to insert teacher assistants of possible intermediate scales between the teacher and the student so that the gap is drawn closer (Zhang et al., 2022). More recently, teachers with student-friendly architectures have exactly showed the significance of student-friendliness (Park et al., 2021a). MetaKD (Zhou et al., 2022) adopts meta-learning to optimize the student-friendliness of the teacher according to the student preference. DKD (Zhao et al., 2022) decouples and amplifies student-friendly knowledge in contrast to others. Distinguished from these student-friendly teachers that are achieved by altering teacher scales, architectures, parameters or knowledge representations, our work, to our best knowledge, is the first one suggesting that teacher parameters can produce both...
student-friendly and student-unfriendly knowledge and aiming to find the sparse teacher with the best student-friendliness.

2.3 Model Pruning

Model pruning is imposed to remove the less expressive parameters for model compression. Previous work applies either structured (Li et al., 2017; Luo et al., 2017; He et al., 2017; Yang et al., 2022) or unstructured pruning (Han et al., 2015; Park et al., 2017; Louizos et al., 2018; Lee et al., 2019) to transformers. Unstructured pruning focuses on pruning parameter-level parameters based on zero-order decisions derived from magnitudes (Gordon et al., 2020) or first-order decisions computed from both gradients and magnitudes (Sanh et al., 2020).

In contrast, structured pruning prunes module-level parameters such like MHA heads (Michel et al., 2019) and FFN layers (Prasanna et al., 2020) guided by the expressive score (Michel et al., 2019). It is noteworthy that while some pruning methods leverage post-training pruning (Hou et al., 2020), others can take advantage of training-time pruning (Xia et al., 2022). Although training-time pruning can result in slightly better performance, it can consume much more time to meet a convergence. Our work mainly exploits structured pruning to obtain sparse teachers, yet also explores the use of unstructured pruning, in a post-training style.

3 Sparse Teacher Trick

Our trick involves three stages in the student learning procedure as shown in Figure 2. First, we distil a trial student from the dense teacher on a specific task (trial distillation). Then, we sparsify the parameters of the dense teacher that are associated with adequately low knowledgeable scores (parameter sparsification). Finally, rewinding is applied, where the student is set to the initialization exactly used in the trial distillation stage and is learned from the sparse teacher during (actual distillation).

3.1 Trial and Actual Distillations

Trial distillation and actual distillation share the same distillation regime. We employ the widely-used logits distillation (Hinton et al., 2015) as the distillation objective, as depicted below:

\[
\begin{align*}
L_{KD} &= -\text{softmax}(z^t / \tau) \log \text{softmax}(z^s / \tau), \\
L_{TK} &= -y \log y^s, \\
L &= L_{KD} + \alpha \cdot L_{TK},
\end{align*}
\]

where \( z^t, z^s \) separately stand for logits of the teacher and student, and \( y^s, y \) separately stand for prediction normalized probabilities of the student and ground-truth one-hot probabilities. Two subscripts KD and TK indicate distillation and task losses respectively. \( \tau \) is a temperature controlling the smoothness the logits (Hinton et al., 2015), and \( \alpha \) is a term balancing two losses.

The trial distillation and actual distillation also reuse the initialization of the student for better convergence, which is known as rewinding technique (Frankle and Carbin, 2019).

3.2 Parameter Sparsification

For parameter sparsification, we design a knowledgeable score, which is essentially an interpolation of the already-proposed expressive score (Molchanov et al., 2017) and our proposed student-friendly score, to measure knowledgeable-ness of each teacher parameter. Thanks to the knowledgeable score, we can safely exclude student-unfriendly parameters without harming expressive parameters too much.

We mainly sparsify the attention heads of MHA blocks and intermediate neurons of FFN blocks in the teacher. Following the literature on structured pruning in a post-training style (Michel et al., 2019; Hou et al., 2020), we attach a set of variables \( \xi^{(i)} \) and \( \nu \) to the attention heads and the intermediate neurons, to record the parameter sensitivities for a specific task through accumulated absolute gradients, as shown below:

\[
\begin{align*}
\text{MHA}^\circ(X) &= \sum_{i=1}^{A} \xi^{(i)} \text{Attn}(X, W^{(i)}_Q, W^{(i)}_K, W^{(i)}_V) W^{(i)}_O, \\
\text{FFN}^\circ(X) &= \text{GELU}(XW_1) \text{diag}(\nu) W_2,
\end{align*}
\]

where \( \xi^{(i)} \equiv 1 \) and \( \nu \equiv 1^{d_t} \). We set the values of the \( \xi^{(i)} \) and \( \nu \) to ones to ensure the functionalities of corresponding heads and neurons are retained.

The implementation is mathematically equivalent to the prevalent first-order taylor expansion of the absolute variation between before and after removing a module (i.e., a head or a neuron) akin to Molchanov et al. (2017). Take the \( i \)-th attention head as an example, its parameter sensitivity can
Figure 2: The overview of STARK. trial distillation distills a trial student from the dense teacher on a specific task. parameter sparsification sparsifies the parameters of the dense teacher that are associated with adequately low knowledgeable scores. actual distillation rewinds the trial distillation by replacing the dense teacher with the proposed sparse teacher.

be written as:

\[
\frac{\partial \mathcal{L}}{\partial \xi^{(i)}} = \left[ \frac{\partial \mathcal{L}}{\partial \xi^{(i)}} \frac{\partial \xi^{(i)} \mathbf{O}^{(i)}}{\partial \xi^{(i)}} \right] = \left[ \frac{\partial \mathcal{L}}{\partial \mathbf{O}^{(i)}} \mathbf{O}^{(i)} \right]
\]

\[
\approx \left| (\mathcal{L}_0 + \frac{\partial \mathcal{L}}{\partial \mathbf{O}^{(i)}}(\mathbf{O}^{(i)} - \mathbf{0}) + \mathbf{r}) - \mathcal{L}_0 \right|
\]

\[
= |\mathcal{L} - \mathcal{L}_0|,
\]

where \( \mathcal{L} \) stands for an arbitrary objective with abuse of notation, and \( \mathbf{O}^{(i)} \) is utilized for \( i \)-th attention head output. \( \mathcal{L}_0 \) actually means \( \mathcal{L}\big|_{\mathbf{O}^{(i)}=\mathbf{0}} \), and \( \mathbf{r} \) represents residuals in taylor expansion.

Note that our trick can be flexibly extended to a training-time style (Xia et al., 2022) or unstructured pruning, which will be discussed in our experiments.

**Expressiveness.** The expressiveness of the teacher is tied to the expressiveness score. A higher expressiveness score indicates that the corresponding parameter has bigger contribution towards the performance. Concretely, the expressiveness scores of the attention heads in MHA and the intermediate neurons in FFN can be depicted as:

\[
P_{\text{head}}^{(i)} = \mathbb{E}_D \left| \frac{\partial \mathcal{L}_{\text{TK}}}{\partial \xi^{(i)}} \right|,
\]

\[
P_{\text{neuron}} = \mathbb{E}_D \left| \frac{\partial \mathcal{L}_{\text{TK}}}{\partial \text{diag}(\nu)} \right|,
\]

where \( D \) is a data distribution, and \( \mathcal{L}_{\text{TK}} \) is the task loss of the teacher. \( \mathbb{E} \) represents expectation.

**Student-friendliness.** Likewise, the student-friendliness of the teacher can be described as student-friendliness scores, which are approximated from distillation loss of the trial distillation.

\[
Q_{\text{head}}^{(i)} = \mathbb{E}_D \left| \frac{\partial \mathcal{L}_{\text{KD}}}{\partial \xi^{(i)}} \right|,
\]

\[
Q_{\text{neuron}} = \mathbb{E}_D \left| \frac{\partial \mathcal{L}_{\text{KD}}}{\partial \text{diag}(\nu)} \right|,
\]

where \( \mathcal{L}_{\text{KD}} \) is the distillation loss as computed with the trial student from the trial distillation. Accordingly, the higher the student-friendliness score is, the more friendliness the teacher offers.

Referring to Molchanov et al. (2017), we normalize the expressiveness and student-friendliness scores with \( \ell_2 \) norm. In view that the teacher needs to balance the expressiveness and student-friendliness, we introduce a coefficient \( \lambda \) to quantify the tradeoff. Therefore, the knowledgeable score can be written in an interpolated form:

\[
P_{\text{head}}^{(i)} = \lambda \cdot P_{\text{head}}^{(i)} + (1 - \lambda) \cdot Q_{\text{head}}^{(i)}
\]

\[
P_{\text{neuron}} = \lambda \cdot P_{\text{neuron}} + (1 - \lambda) \cdot Q_{\text{neuron}}.
\]

**Parameter sparsification** sparsifies the parameters in the teacher with adequately low knowledgeable scores. The adequacy is met by enumerating diverse sparsity levels and obtaining the one leading to the best student during the actual distillation.

4 Experiments

4.1 Data & Metrics

We evaluate our approach on GLUE benchmark (Wang et al., 2019) that contains a collection of NLU tasks, including CoLA (Warstadt et al., 2019) for linguistic acceptability, SST-2 (Socher et al., 2013) for sentiment analysis, MRPC (Dolan and Brockett, 2005), QQP\(^3\) and STS-B (Cer et al., 2017) for paraphrase similarity.

---

\(^3\)https://data.quora.com/

First-Quora-Dataset-Release-Question-Pairs
Dataset | #Train exam. | #Dev exam. | Max. length | Metric
--- | --- | --- | --- | ---
SST-2 | 67K | 0.9K | 64 | Accuracy
MRPC | 3.7K | 0.4K | 128 | F1
STS-B | 7K | 1.5K | 128 | Spearman Correlation
QQP | 364K | 40K | 128 | F1
MNLI-m/mm | 393K | 20K | 128 | Accuracy
QNLI | 105K | 5.5K | 128 | Accuracy
RTE | 2.5K | 0.3K | 128 | Accuracy

Table 1: The statistics, maximum sequence lengths, and metrics of the GLUE benchmark.

matching, MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016) and RTE (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009) for natural language inference. Note that we exclude CoLA (Warstadt et al., 2019) on which general knowledge distillation methods transfer knowledge poorly (Xia et al., 2022).

Accuracy is adopted as the evaluation metric for MNLI-m, MNLI-mm, QNLI, RTE and SST-2, and F1-score is used for MRPC, QQP. The Spearman correlation is used for STS-B. We also report the Average results on development sets of all datasets. We display the statistics of GLUE in Table 1.

4.2 Implementation & Baselines

We conduct experiments on an Nvidia V100. AdamW (Loshchilov and Hutter, 2019) is applied as the optimizer. We search the learning rate within \{1, 2, 3\} \times 10^{-5} and the batch size within \{16, 32\}. All training procedures are carried out within 10 epochs, with an early-stopping. We empirically find that, when temperature \( \tau \) is 2.0 and distillation balance \( \alpha \) is 1.0, reasonable performance is attained. The optimal sparsity is searched within \{10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%\}. Knowledgeableness tradeoff \( \lambda \) is set to 0.5 for acceptable performance and its impact on the performance will be discussed later.

We finetune the original BERT\(^4\) as the teacher and distil it to the student of a smaller scale initialized by dropping 2/3 layers or pruning 70% parameters (with above-mentioned expressiveness pruning) of the teacher, which is initialized from the teacher. We first directly finetune the student as a solid baseline (FT). Then we compare our method to conventional baselines, such as KD (Hinton et al., 2015), PKD (Sun et al., 2019), CKD (Park et al., 2021b), and DynaBERT (Hou et al., 2020). Further, we compare our method to student-friendly baselines, including TAKD that employs a reasonable assistant (Mirzadeh et al., 2020), MetaKD (Zhou et al., 2022) that adapts the teacher with the student feedback, and DKD (Zhao et al., 2022) that amplifies the student-friendly knowledge.

4.3 Main Comparison

Table 2 shows the main experimental results. We can observe that STARK has a significant performance gain by comparing STARK with the original KD. Numerically, the absolute improvements brought by STARK are 1.0% and 0.4% in term of Average. This result implies that sparse teachers can be dense with knowledge. On another note, this possibly indicates a good teacher should be a modest one. Moreover, STARK achieves 0.7% and 0.3% absolute improvements when compared to the competitive TAKD, illustrating that sparse teachers can be more expressive and student-friendly, thereby more knowledgeable to the student than teacher assistants. It seems that student-friendly baselines can only realize a comparable performance to the conventional baselines. We argue this is not the case when student-friendly baselines, say DKD, are armed with advanced distillation objectives, say PKD. Also note that the performances of MetaKD and DynaBERT are lower than those originally reported, as the original work either initialized the student from a pretrained LM of the same scale or utilized extra augmented data.

4.4 Analyses

Knowledgeableness Tradeoff

To investigate the impact of the tradeoff between expressiveness and student-friendliness, we conduct more experiments by varying \( \lambda \) values. Figure 3 illustrates the performance variation along with the change of \( \lambda \). The performance generally exhibits a concave curvature, which hints that the sparsification of the teacher does face a tradeoff between expressiveness

\(^4\)https://github.com/google-research/bert
Table 2: The results of main comparison on GLUE development set. The best results on datasets are boldfaced. § is the optimal sparsity on each dataset. *4 and *30% mean the student is initialized by dropping 2/3 layers or pruning 70% parameters of the teacher. STARK4 and STARK30% exactly mean KD4 and KD30% w/ STARK. We only report MetaKD on small datasets due to limited resources, and DynaBERT without data augmentation due to unavailable augmented data.

<table>
<thead>
<tr>
<th>Method</th>
<th>MNLI-m Acc</th>
<th>MNLI-mm Acc</th>
<th>MRPC F1</th>
<th>QNLI Acc</th>
<th>QQP F1</th>
<th>RTE Acc</th>
<th>STSB SpCorr</th>
<th>SST-2 Acc</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT_base</td>
<td>84.9</td>
<td>84.9</td>
<td>91.2</td>
<td>91.7</td>
<td>88.4</td>
<td>71.5</td>
<td>88.3</td>
<td>93.8</td>
<td>86.8</td>
</tr>
</tbody>
</table>

layer-dropped student

<table>
<thead>
<tr>
<th>Method</th>
<th>MNLI-m Acc</th>
<th>MNLI-mm Acc</th>
<th>MRPC F1</th>
<th>QNLI Acc</th>
<th>QQP F1</th>
<th>RTE Acc</th>
<th>STSB SpCorr</th>
<th>SST-2 Acc</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT4</td>
<td>77.5</td>
<td>77.7</td>
<td>86.0</td>
<td>85.3</td>
<td>86.1</td>
<td>65.0</td>
<td>86.5</td>
<td>89.5</td>
<td>81.7</td>
</tr>
<tr>
<td>KD4</td>
<td>77.7</td>
<td>77.7</td>
<td>86.9</td>
<td>85.1</td>
<td>86.1</td>
<td>65.3</td>
<td>86.4</td>
<td>89.9</td>
<td>82.0</td>
</tr>
<tr>
<td>PKD4</td>
<td>77.7</td>
<td>77.7</td>
<td>87.6</td>
<td>85.0</td>
<td>86.0</td>
<td>65.3</td>
<td>86.4</td>
<td>89.9</td>
<td>82.0</td>
</tr>
<tr>
<td>CKD4</td>
<td>77.7</td>
<td>77.9</td>
<td>87.2</td>
<td>85.0</td>
<td>86.2</td>
<td>64.6</td>
<td>86.4</td>
<td>89.6</td>
<td>81.8</td>
</tr>
<tr>
<td>MetaKD4</td>
<td>\</td>
<td>\</td>
<td>85.1</td>
<td>\</td>
<td>\</td>
<td>63.9</td>
<td>86.5</td>
<td>89.5</td>
<td>\</td>
</tr>
<tr>
<td>DKD4</td>
<td>77.9</td>
<td>78.0</td>
<td>86.9</td>
<td>84.8</td>
<td>86.0</td>
<td>66.3</td>
<td>86.5</td>
<td>88.8</td>
<td>81.9</td>
</tr>
<tr>
<td>TAKD4</td>
<td>77.1</td>
<td>77.3</td>
<td>87.2</td>
<td>84.5</td>
<td>86.3</td>
<td>67.9</td>
<td>86.7</td>
<td>89.9</td>
<td>82.1</td>
</tr>
<tr>
<td>STARK4</td>
<td>78.8</td>
<td>79.0</td>
<td>87.4</td>
<td>85.7</td>
<td>86.5</td>
<td>67.5</td>
<td>87.2</td>
<td>90.6</td>
<td>82.8</td>
</tr>
<tr>
<td>§</td>
<td>40%</td>
<td>50%</td>
<td>50%</td>
<td>30%</td>
<td>60%</td>
<td>40%</td>
<td>50%</td>
<td>46%</td>
<td></td>
</tr>
</tbody>
</table>

parameter-pruned student

<table>
<thead>
<tr>
<th>Method</th>
<th>MNLI-m Acc</th>
<th>MNLI-mm Acc</th>
<th>MRPC F1</th>
<th>QNLI Acc</th>
<th>QQP F1</th>
<th>RTE Acc</th>
<th>STSB SpCorr</th>
<th>SST-2 Acc</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT30%</td>
<td>82.0</td>
<td>82.6</td>
<td>88.5</td>
<td>89.5</td>
<td>87.7</td>
<td>69.0</td>
<td>87.2</td>
<td>91.9</td>
<td>84.8</td>
</tr>
<tr>
<td>KD30%</td>
<td>82.5</td>
<td>82.4</td>
<td>89.1</td>
<td>89.5</td>
<td>87.8</td>
<td>69.3</td>
<td>87.0</td>
<td>91.9</td>
<td>84.9</td>
</tr>
<tr>
<td>PKD30%</td>
<td>82.5</td>
<td>82.8</td>
<td>89.5</td>
<td>89.9</td>
<td>88.0</td>
<td>68.6</td>
<td>86.4</td>
<td>91.9</td>
<td>84.9</td>
</tr>
<tr>
<td>DynaBERT30%</td>
<td>81.5</td>
<td>82.8</td>
<td>87.4</td>
<td>89.1</td>
<td>86.6</td>
<td>68.1</td>
<td>87.2</td>
<td>90.3</td>
<td>84.1</td>
</tr>
<tr>
<td>DKD30%</td>
<td>82.4</td>
<td>82.4</td>
<td>88.4</td>
<td>89.6</td>
<td>87.7</td>
<td>70.4</td>
<td>87.0</td>
<td>91.9</td>
<td>85.0</td>
</tr>
<tr>
<td>TAKD30%</td>
<td>82.7</td>
<td>82.3</td>
<td>89.1</td>
<td>89.8</td>
<td>87.8</td>
<td>68.6</td>
<td>87.6</td>
<td>91.9</td>
<td>85.0</td>
</tr>
<tr>
<td>STARK30%</td>
<td>82.8</td>
<td>82.9</td>
<td>89.4</td>
<td>90.0</td>
<td>87.8</td>
<td>69.7</td>
<td>87.9</td>
<td>92.2</td>
<td>85.3</td>
</tr>
<tr>
<td>§</td>
<td>30%</td>
<td>20%</td>
<td>30%</td>
<td>70%</td>
<td>40%</td>
<td>20%</td>
<td>30%</td>
<td>40%</td>
<td>35%</td>
</tr>
</tbody>
</table>

Figure 3: Performance of STARK4 with different $\lambda$.

Scalability To examine the scalability of STARK to larger teachers (i.e., BERT_large) and smaller students (i.e., *2), where distillation methods can in fact suffer more severely from student-unfriendliness, we distill from BERT_large to an eight-layer student with KD and STARK, and also distill from BERT_base to an two-layer student. The results shown in Table 3 suggest that STARK works well on large teachers and smaller students, and the capacity gap between large teachers and small students can be drawn closer by selecting a sparse teacher. However, the eight-layer student distilled from BERT_large performs only slightly better than the four-layer student distilled from BERT_base even with STARK (see Table 2). With 1/3 parameters, STARK4 can achieve 95% performance of BERT_base, and such 95%/33% scale-performance tradeoff is acceptable in real-world applications. In contrast, the two-layer student can only get a 85%/17% tradeoff, limiting its practical usage.

Training Efficiency STARK indeed requires more training time compared to KD due to the exhaustive search during the actual distillation stage. However, it does not introduce heavy compute since the search mainly involves additional distillations with sparsified teachers that are smaller than the original teacher. Table 4 indicates that actual
Table 3: The results of scalability to larger teachers and smaller students.

<table>
<thead>
<tr>
<th>Method</th>
<th>MNLI-m Acc</th>
<th>MNLI-mm Acc</th>
<th>MRPC F1</th>
<th>QNLI Acc</th>
<th>QQP Acc</th>
<th>RTE Acc</th>
<th>STSB SpCorr</th>
<th>SST-2 Acc</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT_{large}</td>
<td>86.6</td>
<td>86.1</td>
<td>92.3</td>
<td>92.2</td>
<td>89.0</td>
<td>75.5</td>
<td>89.9</td>
<td>93.9</td>
<td>88.2</td>
</tr>
<tr>
<td>KD_8</td>
<td>78.9</td>
<td>79.5</td>
<td>84.9</td>
<td>86.1</td>
<td>86.4</td>
<td>63.9</td>
<td>85.6</td>
<td>90.5</td>
<td>82.0</td>
</tr>
<tr>
<td>STARK_8 §</td>
<td>79.4</td>
<td>80.5</td>
<td>85.0</td>
<td>86.3</td>
<td>87.0</td>
<td>65.7</td>
<td>88.7</td>
<td>90.9</td>
<td>82.9</td>
</tr>
<tr>
<td>BERT_{base}</td>
<td>84.9</td>
<td>84.9</td>
<td>91.2</td>
<td>91.7</td>
<td>88.4</td>
<td>71.5</td>
<td>88.3</td>
<td>93.8</td>
<td>86.8</td>
</tr>
<tr>
<td>KD_2</td>
<td>73.2</td>
<td>72.8</td>
<td>82.9</td>
<td>78.9</td>
<td>83.5</td>
<td>58.5</td>
<td>46.5</td>
<td>86.8</td>
<td>72.9</td>
</tr>
<tr>
<td>STARK_2 §</td>
<td>73.9</td>
<td>74.3</td>
<td>83.1</td>
<td>80.4</td>
<td>83.8</td>
<td>57.8</td>
<td>48.6</td>
<td>88.1</td>
<td>73.7</td>
</tr>
</tbody>
</table>

Table 4: The training time consumed during trial distillation and actual distillation stages.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Train time on MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>trial distillation</td>
<td>~2.5h</td>
</tr>
<tr>
<td>actual distillation</td>
<td>~7h</td>
</tr>
</tbody>
</table>

distillation consumes not that much more training time than trial distillation. Hence, we believe the tradeoff between training time and student performance, along with training efficiency, is acceptable.

Pluggability We also show STARK is pluggable to any distillation methods since it is orthogonal to existing paradigms. We hence plug STARK to our baselines KD, PKD, and CKD to distil a four-layer student from BERT_{base}. As in Table 5, we observe that STARK has universal pluggability to regarded baselines, averagely improving the absolute performance by 0.9%.

Unstructured Pruning As aforementioned, STARK can be flexibly applied with unstructured pruning. For unstructured pruning, we derive the expressiveness and student-friendliness scores in the same way as that used in our structured STARK, except the recording variables are attached to parameters rather than modules like heads. The results in Table 6 verify that STARK with unstructured pruning is slightly worse that STARK with structured pruning, yet it still outperforms KD. Thus, STARK is capable of unstructured pruning.

Automatic STARK An issue with STARK is that the optimal sparsity is obtained by exhaustively enumerating all candidate sparsity levels, leading to some level of training-inefficiency. To address it, we explore an alternative algorithm to get the optimal sparsity so that STARK is enabled with a pursued automatic property. To this end, an attentive solution is proposed based on a surprising observation that a sparse teacher under the guidance of randomness (denoted as STARK-RAND_4) can achieve a promising Average score of 82.5%, whereas the scores for KD_4 and STARK_4 are correspondingly 81.8% and 82.8%. This weird phenomenon drives us to put forward a proposition.

Assumption 1. Both expressiveness and student-friendliness scores are densely located at their clusters, where the cluster center of student-friendliness scores owns a smaller magnitude than that of expressiveness scores.

The assumption is intuitively verified in Figure 4. When random pruning is conducted, firstly the probability of sparsifying a student-unfriendly parameter is high, and secondly the joint probability of sparsifying a student-unfriendly and inexpress-
<table>
<thead>
<tr>
<th>Method</th>
<th>MNLI-m Acc</th>
<th>MNLI-mm Acc</th>
<th>MRPC F1</th>
<th>QNLI Acc</th>
<th>QQP Acc</th>
<th>RTE Acc</th>
<th>STSB SpCorr</th>
<th>SST-2 Acc</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERTbase</td>
<td>84.9</td>
<td>84.9</td>
<td>91.2</td>
<td>91.7</td>
<td>88.4</td>
<td>71.5</td>
<td>88.3</td>
<td>93.8</td>
<td>86.8</td>
</tr>
<tr>
<td>KD4</td>
<td>77.7</td>
<td>77.7</td>
<td>86.9</td>
<td>85.1</td>
<td>86.1</td>
<td>65.3</td>
<td>86.4</td>
<td>89.6</td>
<td>81.8</td>
</tr>
<tr>
<td>w/ STARK</td>
<td>78.8</td>
<td>79.0</td>
<td>87.4</td>
<td>85.7</td>
<td>86.5</td>
<td>67.5</td>
<td>87.2</td>
<td>90.6</td>
<td>82.8</td>
</tr>
<tr>
<td>PKD4</td>
<td>77.7</td>
<td>77.7</td>
<td>87.6</td>
<td>85.0</td>
<td>86.0</td>
<td>65.3</td>
<td>86.4</td>
<td>89.9</td>
<td>82.0</td>
</tr>
<tr>
<td>w/ STARK</td>
<td>78.8</td>
<td>79.1</td>
<td>87.7</td>
<td>85.9</td>
<td>86.6</td>
<td>66.8</td>
<td>87.2</td>
<td>90.1</td>
<td>82.8</td>
</tr>
<tr>
<td>CKD4</td>
<td>77.7</td>
<td>77.9</td>
<td>87.2</td>
<td>85.0</td>
<td>86.2</td>
<td>64.6</td>
<td>86.4</td>
<td>89.6</td>
<td>81.8</td>
</tr>
<tr>
<td>w/ STARK</td>
<td>78.8</td>
<td>79.0</td>
<td>87.6</td>
<td>86.4</td>
<td>86.5</td>
<td>66.4</td>
<td>87.2</td>
<td>90.4</td>
<td>82.8</td>
</tr>
</tbody>
</table>

Table 5: The results of pluggability to baselines.

<table>
<thead>
<tr>
<th>Method</th>
<th>MNLI-m Acc</th>
<th>MNLI-mm Acc</th>
<th>MRPC F1</th>
<th>QNLI Acc</th>
<th>QQP Acc</th>
<th>RTE Acc</th>
<th>STSB SpCorr</th>
<th>SST-2 Acc</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERTbase</td>
<td>84.9</td>
<td>84.9</td>
<td>91.2</td>
<td>91.7</td>
<td>88.4</td>
<td>71.5</td>
<td>88.3</td>
<td>93.8</td>
<td>86.8</td>
</tr>
<tr>
<td>KD4</td>
<td>77.7</td>
<td>77.7</td>
<td>86.9</td>
<td>85.1</td>
<td>86.1</td>
<td>65.3</td>
<td>86.4</td>
<td>89.6</td>
<td>81.8</td>
</tr>
<tr>
<td>STARK4</td>
<td>78.8</td>
<td>79.0</td>
<td>87.4</td>
<td>85.7</td>
<td>86.5</td>
<td>67.5</td>
<td>87.2</td>
<td>90.6</td>
<td>82.8</td>
</tr>
<tr>
<td>STARK4*</td>
<td>79.0</td>
<td>79.0</td>
<td>87.4</td>
<td>85.3</td>
<td>86.8</td>
<td>66.1</td>
<td>87.3</td>
<td>89.8</td>
<td>82.6</td>
</tr>
</tbody>
</table>

Table 6: The results of compatibility with unstructured pruning. * indicates that unstructured pruning is otherwise used.

A follow-up observation is that STARK-RAND4 expects a smaller average sparsity (25%) than STARK4 does (46%). It is easy to understand that the more parameters are sparsified, the lower probability above the performance guarantee will hold, though. The evident phenomenon inspires us to make another assumption.

**Assumption 2.** An optimal sparsity is positively correlated to the first density peak of a sparsification sequence.

The assumption is illustrated in Figure 5. Since STARK-RAND sparsifies parameters at random, it will have a small optimal sparsity as a consequence of meeting the first density peak very early. For STARK enjoys a sparsification sequence with only one density peak, its optimal sparsity can be automatically estimated (denoted as STARK-AUTO) through Assumption 2. Experimental results can be found in Table 7, where STARK-AUTO approximates STARK in term of the Average metric. Nevertheless, we argue it is the last to use STARK-AUTO otherwise for an extremely low practical compute as the performance can suffer a subtle drop.

5 Conclusions

In this paper, we validate that sparse teachers can be dense with knowledge under the guidance of our designed knowledgeable score. The idea of the sparse teacher is motivated from a pilot study, and the knowledgeable score is carefully crafted to make sure that the student-unfriendly knowl-
edge can be reduced without hurting too much the expressive knowledge. Extensive experimental results on the GLUE benchmark support our claim to a large degree.

**Limitations**

STARK can be further explored under two additional settings: 1) in a task-agnostic setting (e.g., MiniLM) and 2) on large LMs (e.g., BERT\textsubscript{large}). Moreover, our attentive automatic solution for STARK can be enhanced so that its performance can at least match the original performance.

**Acknowledgements**

We thank the anonymous reviewers and chairs for their constructive suggestions. This research was supported in part by Natural Science Foundation of Beijing (grant number: 4222036) and Huawei Technologies (grant number: TC20201228005).

**References**


<table>
<thead>
<tr>
<th>Method</th>
<th>MNLI-m Acc</th>
<th>MNLI-mm Acc</th>
<th>MRPC F1</th>
<th>QNLI Acc</th>
<th>QQP Acc</th>
<th>RTE Acc</th>
<th>STSB SpCorr</th>
<th>SST-2 Acc</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARK\textsubscript{4}</td>
<td>78.8</td>
<td>79.0</td>
<td>87.4</td>
<td>85.7</td>
<td>86.5</td>
<td>67.5</td>
<td>87.2</td>
<td>90.6</td>
<td>82.8</td>
</tr>
<tr>
<td>§</td>
<td>40%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>30%</td>
<td>60%</td>
<td>40%</td>
<td>30%</td>
<td>46%</td>
</tr>
<tr>
<td>STARK-AUTO\textsubscript{4}</td>
<td>78.1</td>
<td>79.0</td>
<td>86.6</td>
<td>85.7</td>
<td>86.0</td>
<td>67.5</td>
<td>87.2</td>
<td>90.0</td>
<td>82.6</td>
</tr>
<tr>
<td>§</td>
<td>47%</td>
<td>51%</td>
<td>35%</td>
<td>46%</td>
<td>44%</td>
<td>42%</td>
<td>38%</td>
<td>38%</td>
<td>43%</td>
</tr>
</tbody>
</table>

Table 7: The results of STARK-AUTO.


Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. 2017. When BERT plays the lottery, all tickets are winning. In ICLR.


Jongsoo Park, Sheng R. Li, Wei Wen, Ping Tak Peter Tang, Hai Li, Yiran Chen, and Pradeep Dubey. 2017. Faster cnns with direct sparse convolutions and guided pruning. In ICLR.


Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT plays the lottery, all tickets are winning. In EMNLP, pages 3208–3229.


Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In EMNLP, pages 1631–1642.


A  Comparative Equivalence of Distribution Variance and Negative Entropy

Theorem 1. For any two distributions $y$ and $y'$, the negative entropy difference between them can be approximated by their variance difference.

Proof.

$$-H(y) - (-H(y'))$$
$$= \sum_i y_i \log y_i - \sum_i y'_i \log y'_i$$
$$\approx \sum_i (y_i - 1) + \frac{1}{2} (y_i - 1)^2 + r$$
$$- \sum_i (y'_i - 1) + \frac{1}{2} (y'_i - 1)^2 + r$$
$$= \sum_i -(y_i - y'_i) + \frac{1}{2} (y_i^2 - y_i'^2)$$
$$= \sum_i \frac{1}{2}((y_i - \bar{y})^2 - (y'_i - \bar{y'})^2)$$
$$\propto \sum_i (y_i - \bar{y})^2 - \sum_i (y'_i - \bar{y'})^2$$
$$= V(y) - V(y').$$

Corollary 1. Distribution variance, when taken as the measure of confidence, is comparatively equivalent to distribution negative entropy.

B  Density of Scores of BERT$_{\text{base}}$ Intermediate Neurons on MRPC

C  Density of Scores of BERT$_{\text{base}}$ Attention Heads on GLUE
Figure 8: Density and cumulative density of knowledgeableness scores of BERT\textsubscript{base} attention heads finetuned on GLUE.