Multi-Granularity Structural Knowledge Distillation for Language Model Compression

Chang Liu[1,2], **Chongyang Tao**[3]*, **Jiazhan Feng**[1], **Dongyan Zhao**[1,2,4,5]*
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Center for Data Science, Peking University
[3]Microsoft Corporation
[4]Artificial Intelligence Institute of Peking University
[5]State Key Laboratory of Media Convergence Production Technology and Systems
{liuchang97,fengjiazhan,zhaody}@pku.edu.cn,
chotao@microsoft.com

Abstract

Transferring the knowledge to a small model through distillation has raised great interest in recent years. Prevailing methods transfer the knowledge derived from mono-granularity language units (e.g., token-level or sample-level), which is not enough to represent the rich semantics of a text and may lose some vital knowledge. Besides, these methods form the knowledge as individual representations or their simple dependencies, neglecting abundant structural relations among intermediate representations. To overcome the problems, we present a novel knowledge distillation framework that gathers intermediate representations from multiple semantic granularities (e.g., tokens, spans and samples) and forms the knowledge as more sophisticated structural relations specified as the pair-wise interactions and the triplet-wise geometric angles based on multi-granularity representations. Moreover, we propose distilling the well-organized multi-granularity structural knowledge to the student hierarchically across layers. Experimental results on GLUE benchmark demonstrate that our method outperforms advanced distillation methods.

1 Introduction

Recent years have witnessed a surge of pre-trained language models (Devlin et al., 2019; Lewis et al., 2020; Clark et al., 2020; Brown et al., 2020). Building upon the transformer architecture (Vaswani et al., 2017) and pre-trained on large-scale corpora using self-supervised objectives, these PLMs have achieved remarkable success in a wide range of natural language understanding and generation tasks. Despite their high performance, these PLMs usually suffer from high computation and memory costs, which hinders them from being deployed into resource-scarce scenarios, e.g., mobile phones and embedded devices.

Various attempts have been made to compress the huge PLMs into small ones with minimum performance degradation. As one of the main approaches, knowledge distillation (Hinton et al., 2015) utilizes a large and powerful teacher model to transfer the knowledge to a small student model. Based on the teacher-student framework, Jiao et al. (2020); Wang et al. (2020) distilled the token-level representations and attention dependencies to the student, Sanh et al. (2019); Sun et al. (2019) taught the student to mimic the output logits of the teacher, Sun et al. (2020) enforced the student's representation to be closed to the teacher's while pushing negative samples to be far apart. Although proved effective, existing approaches have some flaws. For one thing, these distillation methods only adopted the representations of mono-granularity language units (i.e., token-level or sample-level), while neglecting other granularity. For another, their distillation objectives either matched the corresponding representations between the teacher and the student or aligned the attention dependencies, failing to capture more sophisticated structural relations between the representations.

To address these issues, in this paper we propose a novel knowledge distillation framework named **M**ulti-**G**ranularity **S**tructural **K**nowledge **D**istillation (MGSKD) through answering the three research questions: (1) *which* granularity should the knowledge be, (2) *what* form of knowledge is effective to transfer and (3) *how* to teach the student using the knowledge. For the "*which*" question, given that natural languages have multiple semantic granularities, we consider the intermediate representations in three granularities: tokens, spans and samples. Specifically, we first take the sub-word tokens as the smallest granularity, then

*Corresponding authors: Chongyang Tao and Dongyan Zhao.

select phrases and whole words as spans for they hold complete meanings, and finally treat the whole input texts as samples. We use mean-pooling to obtain the representations of spans and samples based on token representations. For the "*what*" question, we propose to leverage the sophisticated structural relations between the representations as the knowledge. Concretely, instead of aligning the corresponding representations of the teacher and the student, we propose to form the knowledge as the pair-wise interactions and the triplet-wise geometric angels of a group of representations. For the "*how*" question, following the recent findings that the bottom layers capture syntactic features while the upper layers encode semantic features (Jawahar et al., 2019), we conduct hierarchical distillation where the bottom layers of the student are taught token-level and span-level knowledge while the upper layers learn sample-level knowledge.

We conduct comprehensive experiments on standard language understanding benchmark GLUE (Wang et al., 2018). Experimental results demonstrate that our knowledge distillation framework outperforms strong baselines methods. Surprisingly, MGSKD achieves comparable or better performance than BERT$_{\text{base}}$ on most of the tasks on GLUE, while keeping much smaller and faster. Our contributions in this paper are three folds:

- We are the first to leverage multi-granularity semantic representations in language (i.e., the representations of tokens, spans and samples) for knowledge distillation.
- We propose to form the knowledge as sophisticated structural relations specified as the pair-wise interactions and the triplet-wise geometric angles based on multi-granularity representations.
- We conduct comprehensive experiments on GLUE benchmark and MGSKD achieves superior results over other knowledge distillation baselines.

2 Related Work

Language Model Compression. Pre-trained language models (Devlin et al., 2019; Clark et al., 2020; Brown et al., 2020) perform remarkably well on various applications but at the cost of high computation and memory usage. To deploy these powerful models into resource-scarce scenarios, various attempts have been made to compress the language models into small ones. Quantization methods (Zafrir et al., 2019; Shen et al., 2020; Zhang et al., 2020; Bai et al., 2021) convert the model parameters to lower precision. Pruning approaches identify then remove unimportant individual weights or structures (Michel et al., 2019; Fan et al., 2019; Gordon et al., 2020; Hou et al., 2020). Weight sharing techniques (Dehghani et al., 2018; Lan et al., 2019) allow the model to reuse the transformer layer multiple times to reduce parameters.

Knowledge Distillation. Knowledge distillation (Hinton et al., 2015) is another major line of research to do model compression, which is the main concentration in this paper. Hinton et al. (2015) first proposed to minimize the KL-divergence between the predicted distributions of the teacher and the student. Sanh et al. (2019); Sun et al. (2019); Liang et al. (2020) adopted this objective to teach the student on masked language modeling or text classification tasks. Romero et al. (2014) proposed to directly match the feature activations of the teacher and the student. Jiao et al. (2020) followed the idea and took the intermediate representations in each transformer layer of the teacher as one of the knowledge to be transferred. Tian et al. (2019) proposed a contrastive distillation framework where the teacher's representations were treated as positives to the corresponding student's representations. Sun et al. (2020); Fu et al. (2021) customized this idea to language model compression and proved its effectiveness. Researchers also attempted to use the mutual relations of representations as the knowledge to transfer. In the literature of image classification, Peng et al. (2019); Tung and Mori (2019); Park et al. (2019) pointed out that the relations of the image representations of the teacher should be preserved in the student's feature space, and adopted a series of geometric measurements to model the sample relations. For distilling transformer models, Park et al. (2021) enforced the relations across tokens and layers between the teacher and the student to be consistent. Jiao et al. (2020); Wang et al. (2020, 2021) used the attention dependencies between tokens to teach the student. In this paper, we propose to transfer the multi-granularity knowledge to the student. Different from previous works that only considered a single granularity of representations, we jointly transfer the token-level, span-level and sample-level structural knowledge. And compared with Shao and Chen (2021) which considered the multi-granularity visual features in an image as the knowledge, our method works in a different modality, presents a different definition of granularity,

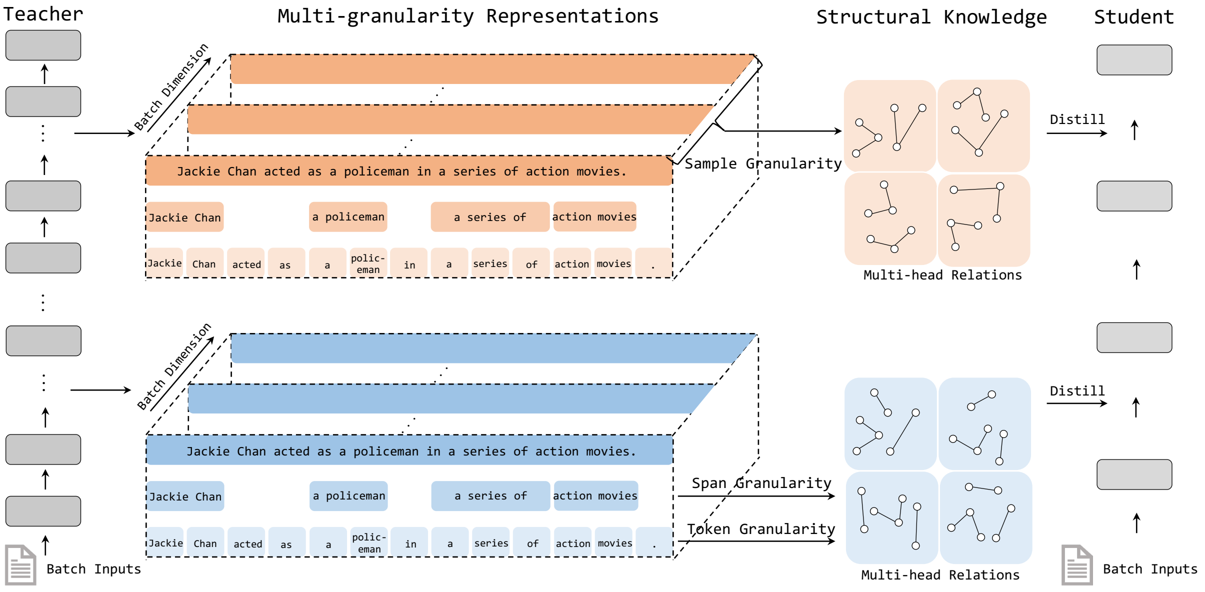


Figure 1: The overall framework of MGSKD.

and prepares the multi-granularity knowledge as the structural relations among representations.

3 Method

We propose **M**ulti-**G**ranularity **S**tructural **K**nowledge **D**istillation, a novel framework to distill the knowledge from a large transformer language model to a small one. Different from previous works that transferred the knowledge derived from either token-level or sample-level outputs, we prepare the knowledge in three semantic granularities: token-level, span-level and sample-level. Given some granularity of representations of the teacher model, we form the knowledge as the structural relations, i.e., the pair-wise interactions and the triplet-wise geometric angles, between the representations. We then distill the well-organized structural knowledge to the student hierarchically across layers, where the token-level and the span-level knowledge are transferred to the bottom layers to provide more syntactic guidance while the sample-level knowledge is transferred to the upper layers to offer more help of semantic understanding. The framework of MGSKD is illustrated in Figure 1.

3.1 Multi-granularity Representation

Natural languages have multiple granularities of conceptual units. In the context of pre-trained transformers (Devlin et al., 2019), the basic unit is the tokens produced by sub-word tokenizers (Wu et al., 2016; Radford et al., 2019). Several consecutive tokens become a text span, and the sample is comprised of all the tokens it contains. Existing knowledge distillation approaches (Jiao et al., 2020; Wang et al., 2020; Sun et al., 2020; Fu et al., 2021) focused on one granularity of representation, neglecting that texts are built upon language units from multiple granularities. Intuitively, incorporating multi-granularity representations in knowledge distillation may provide more guidance since the student can be taught how to compose the semantic concepts from small granularities to larger ones. Therefore, we propose to gather multi-granularity representations for knowledge distillation. We construct three granularities of representations: tokens, spans that hold complete meanings, and samples.

Token Representation. The first granularity is the sub-word token, which is the foundation of high-level granularity. Given an input text, a tokenizer such as WordPiece (Wu et al., 2016) splits it into n tokens $x = [t_1, t_2, \ldots, t_n]$. The tokens are converted to a sequence of continuous representations $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n] \in \mathbb{R}^{n\times d}$ through the embedding layer. For the sake of clarity, we treat the embedding layer as the 0-th layer and set $\boldsymbol{H}^0 = \boldsymbol{E}$. Then the token embeddings $\boldsymbol{H}^0$ are passed to L stacked transformer layers. The l-th layer takes the output representations $\boldsymbol{H}^{l-1}$ of the previous layer as its input, and returns the updated representations $\boldsymbol{H}^l$ using multi-head attention (MHA) and position-wise feed-forward network (FFN). Herein, we obtain $L+1$ layers of token representations $\{\boldsymbol{H}^l\}_{l=0}^L$ where $\boldsymbol{H}^l \in \mathbb{R}^{n\times d}$.

Span Representation. The second granularity is the span, which is comprised of several consecutive tokens. Different from SpanBERT (Joshi et al., 2020) that randomly selects token spans whose start positions and lengths are sampled from some distributions for masked language modeling, we propose to extract spans that have complete meanings. Widely adopted sub-word tokenizers in pre-trained transformers split some of the English words into several sub-word tokens. We consider these whole words consisting of multiple sub-word tokens, and phrases, as meaningful spans. Sub-word tokens for whole words are easy to obtain using WordPiece tokenizer (Wu et al., 2016). While for phrase identification, we train a classifier-based English chunker on CoNLL-2000 corpus (Tjong Kim Sang and Buchholz, 2000) following the instructions[1]. We then use the trained chunker to extract noun phrases (NP), verb phrases (VP), and prepositional phrases (PP). These identified phrases are tokenized by WordPiece tokenizer to obtain tokens. Herein, we can obtain n_s token spans $x_{\text{span}} = [s_1, s_2, \ldots, s_{n_s}]$, where $s_i = [t_j, t_{j+1}, \ldots, t_{j+n_{s_i}-1}]$ denotes the i-th span that starts at the j-th token and contains n_{s_i} tokens. We then build span representations based on token representations using mean pooling:

$$\hat{\boldsymbol{h}}_{\boldsymbol{i}}^{\boldsymbol{l}} = \text{Pool}(\boldsymbol{H}^l_{j:j+n_{s_i}}), \tag{1}$$

where $\hat{\boldsymbol{h}}_{\boldsymbol{i}}^{\boldsymbol{l}} \in \mathbb{R}^d$ is the representation of the i-th span in layer l. We obtain $L+1$ layers of span representations as $\{\hat{\boldsymbol{H}}^l\}_{l=0}^L$ where $\hat{\boldsymbol{H}}^l \in \mathbb{R}^{n_s\times d}$.

Sample Representation. The third granularity is the input text sample itself. Based on token representations again, we use mean-pooling to aggregate all the token representations in a text sample to form sample representation:

$$\tilde{\boldsymbol{h}}^{\boldsymbol{l}} = \text{Pool}(\boldsymbol{H}^l), \tag{2}$$

Herein, we get $L+1$ layers of sample representations as $\{\tilde{\boldsymbol{h}}^l\}_{l=0}^L$ where $\tilde{\boldsymbol{h}}^l \in \mathbb{R}^d$.

3.2 Structural Knowledge Extraction

With multi-granularity representations, we then need to formulate the specific knowledge we aim to transfer from the teacher to the student. Considering that an element holds its meaning only when it is put into a semantic space where it has various relations to other elements, we propose that the knowledge is better specified as the structural relations of the representations in a semantic space, instead of the individual representations themselves. Therefore, instead of directly matching each hidden representation between the teacher and the student, we propose to extract structural relations from multi-granularity representations as the knowledge to teach the student. We first project the representations into multiple sub-spaces, then we extract two types of structural knowledge: pair-wise interactions and triplet-wise geometric angles.

Multi-head Modeling. A recent study by Wang et al. (2021) pointed out that distilling knowledge with multiple relation heads helps the student learn better. Therefore, before extracting structural knowledge for intermediate representations, we first project them into m sub-spaces, which we call multi-head modeling. Specifically, given a set of n representations $\boldsymbol{R} \in \mathbb{R}^{n\times d}$, we linearly project them into m sub-spaces whose dimensions are d/m. [2] We use $\boldsymbol{R}' \in \mathbb{R}^{m\times n\times d/m}$ to denote the multi-head representations which are then used for extracting structural knowledge.

Pair-wise Interaction. Given two vectors $\boldsymbol{r}_i, \boldsymbol{r}_j \in \mathbb{R}^{d/m}$ in a sub-space, we calculate their interaction as their scaled dot product:

$$\varphi(\boldsymbol{r}_i, \boldsymbol{r}_j) = \frac{\boldsymbol{r}_i \cdot \boldsymbol{r}_j^\intercal}{\sqrt{d/m}}. \tag{3}$$

Herein, we obtain the multi-head pair-wise interaction features for each pair as $\boldsymbol{P} \in \mathbb{R}^{m\times n\times n}$, where $\boldsymbol{P}_{h,i,j}$ denotes the interaction between the i-th representation and the j-th representation in the sub-space of the h-th relation head. Note that $\boldsymbol{P}$ can be considered as the unnormalized self-attention (Vaswani et al., 2017) scores for the given representations, the difference lies in that in our calculation the queries are identical to the keys.

Triplet-wise Geometric Angle. Pair-wise interaction features only consider two vectors at once, which is not enough to represent the complicated structural relations between representations in the high-dimensional space. Therefore, we propose to model the high-order relations as the geometric angles for triplets of vectors. Specifically, given

[1] https://www.nltk.org/book/ch07.html

[2] For the student model, its representations are linearly projected into intermediate states whose dimensions are the same as the teacher model's hidden dimensions, so that it can be split into m sub-spaces as the teacher model.

a triplet of representations $\boldsymbol{r}_i, \boldsymbol{r}_j, \boldsymbol{r}_k \in \mathbb{R}^{d/m}$, we calculate their geometric angle as:

$$\begin{aligned}\psi(\boldsymbol{r}_i, \boldsymbol{r}_j, \boldsymbol{r}_k) &= cos\angle \boldsymbol{r}_i\boldsymbol{r}_j\boldsymbol{r}_k = \langle \boldsymbol{r}_{ij}, \boldsymbol{r}_{kj} \rangle \\ \boldsymbol{r}_{ij} &= \frac{\boldsymbol{r}_i - \boldsymbol{r}_j}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|_2}, \boldsymbol{r}_{kj} = \frac{\boldsymbol{r}_k - \boldsymbol{r}_j}{\|\boldsymbol{r}_k - \boldsymbol{r}_j\|_2}.\end{aligned} \tag{4}$$

We can calculate the geometric angles for all the triplets, and obtain $\boldsymbol{T} \in \mathbb{R}^{m \times n \times n \times n}$ where $\boldsymbol{T}_{h,i,j,k}$ stands for the angle of $\angle \boldsymbol{r}_i\boldsymbol{r}_j\boldsymbol{r}_k$ in the sub-space of the h-th relation head. As the computation complexity increases cubically with n, such a calculation is infeasible when the number of representations is large. Hereby, we propose a two-stage selection strategy to sequentially select important representations to form angles. Similar to Goyal et al. (2020), we assume that the more attention a representation receives from others, the more important it is. Therefore, we first calculate the self-attention distributions $\boldsymbol{A} \in \mathbb{R}^{m \times n \times n}$ by applying *softmax* function on the last dimension of $\boldsymbol{P}$. Then for the j-th representation, we calculate a global salient score s_j by summing up self-attention distributions across all heads and all queries. Based on the score, we pick the top-k_1 salient representations as vertices. Next, if the i-th representation is selected as vertex, we pick k_2 representations with the highest local salient score to form angles with the vertex. We define the local salient score $s_{i,j}$ as the attention posed by the i-th representation on the j-th representation, The salient scores s_i and $s_{i,j}$ are calculated as follows:

$$s_j = \sum_{h=1}^{m}\sum_{i=1}^{n} \boldsymbol{A}_{h,i,j}, \quad s_{i,j} = \sum_{h=1}^{m} \boldsymbol{A}_{h,i,j}. \tag{5}$$

Therefore, by sequentially selecting salient representations to form angles, we reduce the computation complexity from $\mathcal{O}(mn^3)$ to $\mathcal{O}(mk_1k_2^2)$. By choosing proper k_2 and k_2, we can facilitate the computation of triplet-wise geometric angles for any number of representations.

3.3 Hierarchical Distillation

We utilize the structural knowledge extraction approach described in Sec. 3.2 to prepare knowledge based on three granularities of representations presented in Sec. 3.1 for distillation. Based on the findings that the bottom layers capture syntactic features while the upper layers encode semantic features (Jawahar et al., 2019), we propose to conduct hierarchical distillation for the student where different granularities of knowledge are transferred to different layers. For a teacher model with L_t layers and a student model with L_s layers, we first define a layer mapping function $g(\cdot)$ that maps each student layer to a teacher layer that it learns from. Following previous work (Jiao et al., 2020), we adopt the "uniform strategy" for $g(\cdot)$. Then we transfer token-level and span-level knowledge to the bottom-M layers of the student, while leveraging sample-level knowledge to teach its upper $L_s + 1 - M$ layers.

Token- and Span-level. Specifically, given the token-level and the span-level representations of the teacher $\{\boldsymbol{H}_{\boldsymbol{t}}^l, \hat{\boldsymbol{H}}_{\boldsymbol{t}}^l\}_{l=0}^{L_t}$, we use Eq. 3 and Eq. 4 to calculate the pair-wise interactions and the triplet-wise geometric angles among tokens and spans within a single sample as $\{\boldsymbol{P}_{\boldsymbol{t}}^l, \hat{\boldsymbol{P}}_{\boldsymbol{t}}^l\}_{l=0}^{L_t}$ and $\{\boldsymbol{T}_{\boldsymbol{t}}^l, \hat{\boldsymbol{T}}_{\boldsymbol{t}}^l\}_{l=0}^{L_t}$. Similarly, we can obtain the structural relations of the students: $\{\boldsymbol{P}_{\boldsymbol{s}}^l, \hat{\boldsymbol{P}}_{\boldsymbol{s}}^l\}_{l=0}^{L_s}$ and $\{\boldsymbol{T}_{\boldsymbol{s}}^l, \hat{\boldsymbol{T}}_{\boldsymbol{s}}^l\}_{l=0}^{L_s}$. We then teach the student by minimizing the differences of the structural relations among their representations between the teacher and the student:

$$\begin{aligned}\mathcal{L}_{token} &= \sum_{0 \le l < M} (\ell_1(\boldsymbol{P}_{\boldsymbol{t}}^{g(l)}, \boldsymbol{P}_{\boldsymbol{s}}^l) + \ell_2(\boldsymbol{T}_{\boldsymbol{t}}^{g(l)}, \boldsymbol{T}_{\boldsymbol{s}}^l)) \\ \mathcal{L}_{span} &= \sum_{0 \le l < M} (\ell_1(\hat{\boldsymbol{P}}_{\boldsymbol{t}}^{g(l)}, \hat{\boldsymbol{P}}_{\boldsymbol{s}}^l) + \ell_2(\hat{\boldsymbol{T}}_{\boldsymbol{t}}^{g(l)}, \hat{\boldsymbol{T}}_{\boldsymbol{s}}^l)).\end{aligned} \tag{6}$$

Sample-level. Recall that we obtain $\{\tilde{\boldsymbol{h}}_{\boldsymbol{t}}^{\boldsymbol{l}}\}_{l=0}^{L_t}$ and $\{\tilde{\boldsymbol{h}}_{\boldsymbol{s}}^{\boldsymbol{l}}\}_{l=0}^{L_s}$ for the teacher and the student where $\tilde{\boldsymbol{h}}_{\boldsymbol{t}}^{\boldsymbol{l}}, \tilde{\boldsymbol{h}}_{\boldsymbol{s}}^{\boldsymbol{l}} \in \mathbb{R}^d$. Different from the structural knowledge of tokens and spans which is modeled within a sample, the sample-level structural relations rely on a group of sample representations. Although the choice of samples may make a difference to the overall performance, here we simply gather all the sample representations in a mini-batch to calculate their structural relations as the sample-level knowledge. Specifically, we only focus on the triplet-wise relations $\{\tilde{\boldsymbol{T}}_{\boldsymbol{t}}^{\boldsymbol{l}}\}_{l=0}^{L_t}$ and $\{\tilde{\boldsymbol{T}}_{\boldsymbol{s}}^{\boldsymbol{l}}\}_{l=0}^{L_s}$:

$$\mathcal{L}_{sample} = \sum_{M \le l \le L_s} \ell_2(\tilde{\boldsymbol{T}}_{\boldsymbol{t}}^{g(l)}, \tilde{\boldsymbol{T}}_{\boldsymbol{s}}^l). \tag{7}$$

ℓ_1 and ℓ_2 in Eq. 6 and Eq. 7 are loss functions that measure the distance between the structural relations of the teacher's and the student's representations. We empirically choose MSE for ℓ_1 and Huber loss ($\delta = 1$) for ℓ_2.

Model	#Params	Speedup	SST-2	MRPC	RTE	STS-B	MNLI-(m/mm)	QNLI	QQP	CoLA
			(Acc)	(F1)	(Acc)	(Spear)	(Acc)	(Acc)	(Acc)	(Mcc)
BERT$_{\text{base}}$	109M	×1.0	92.8	90.3	65.3	88.4	84.6/84.4	91.3	91.2	56.8
ELECTRA$_{\text{base}}$	109M	×1.0	95.5	92.7	83.4	91.0	88.8/88.7	93.2	92.0	69.6
DistilBERT	66M	×2.0	91.3	-	59.9	86.9	82.2/ -	89.2	88.5	51.3
MiniLMv2	66M	×2.0	92.4	-	72.1	-	84.2/ -	90.8	91.1	52.5
CKD	66M	×2.0	93.0	89.6	67.3	89.0	83.6/84.1	90.5	91.2	55.1
Student$_{\text{ft}}$	14M	×9.4	89.7	88.0	63.7	84.6	80.2/79.8	86.0	86.9	0.0
Student$^{\dagger}_{\text{MiniLMv2}}$	14M	×9.4	92.9	90.3	67.1	88.7	83.7/83.4	89.5	90.9	43.5
Student$^{\dagger}_{\text{CKD}}$	14M	×9.4	92.8	89.9	66.8	88.7	83.2/82.7	89.3	90.3	**46.4**
Student$^{\dagger}_{\text{MGSKD}}$	14M	×9.4	**93.7**	**90.7**	**67.9**	**89.2**	**84.7/84.3**	**89.6**	**91.6**	44.8

Table 1: Evaluation results on the dev set of GLUE benchmark. The results of the models with 66M parameters are taken from published papers. Our results are averaged for 3 runs with different random seeds. The best results of the student models are in-bold. † means the method is implemented with the same distillation setting as ours.

Overall Objectives. The overall distillation objective for multi-granularity structural knowledge distillation is:

$$\mathcal{L}_1 = \lambda_1 \mathcal{L}_{sample} + \lambda_2 \mathcal{L}_{token} + \lambda_3 \mathcal{L}_{span}, \quad (8)$$

where λ_1, λ_2 and λ_3 are weights of loss functions of different granularities.

After this, we also teach the student to match the prediction distributions with the teacher's for text classification tasks:

$$\mathcal{L}_2 = \tau^2 D_{\text{KL}}(\boldsymbol{z}_t/\tau \| \boldsymbol{z}_s/\tau), \quad (9)$$

where $\boldsymbol{z}_t$ and $\boldsymbol{z}_s$ are the predicted probability distributions of the teacher and the student respectively, τ denotes the temperature.

4 Experiments

4.1 Datasets and Metrics

We conduct our experiments on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018). Sepcifically, there are 2 single-sentence tasks: SST-2 (Socher et al., 2013), CoLA (Warstadt et al., 2019), 3 similarity and paraphrase tasks: MRPC (Dolan and Brockett, 2005), STS-B (Cer et al., 2017), QQP (Chen et al., 2018), and 4 inference tasks: MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), RTE (Bentivogli et al., 2009), WNLI (Levesque et al., 2012). Following previous work (Jiao et al., 2020; Wang et al., 2021; Park et al., 2021), we evaluate our method on 8 datasets except WNLI. We report accuracy on 5 datasets: SST-2, QQP, MNLI, QNLI and RTE. We report F1 score on MRPC, Matthews correlation coefficient on CoLA, and Spearman's rank correlation coefficient on STS-B.

4.2 Implementation Details

We focus on task-specific distillation. We follow Jiao et al. (2020) to augment the training sets for each of the GLUE tasks using the code[3] they released. We fine-tune ELECTRA$_{\text{base}}$ on the original training sets as the teacher model, and utilize TinyBERT-4-312[4] which is distilled on general corpora as the initialization of our student model. For token-level and span-level distillation, we use 64 relation heads for calculating pair-wise interactions, and 1 relation head for triplet-wise angles due to its huge computation and memory costs. And we set $k_1 = k_2 = 20$ for calculating angles. For sample-level distillation, we use 64 relation heads and set k_1 and k_2 as the batch size. We distill token-level and span-level knowledge to the bottom-2 layers of the student and distill sample-level knowledge to the other layers. For the structural distillation objective, we set $\lambda_1 = 4$, $\lambda_2 = \lambda_3 = 1$ to maintain their gradient norms in the same order of magnitude. We first distill the student model using Eq. 8 for 50 epochs on CoLA and 20 epochs on other tasks. The learning rate is 1e-5 and the batch size is 32. Then we use Eq. 9 to distill the predictions for all tasks except STS-B since we empirically find that directly fine-tuning after distillation using Eq. 8 yields better performance for it. For QQP and CoLA, we adopt the original training set and distill the student for 10 epochs while for other 5 tasks we use the augmented training sets and distill the student for 3 epochs. We set τ as 1.0, the learning

[3] https://github.com/huawei-noah/Pretrained-Language-Model/blob/master/TinyBERT/data_augmentation.py

[4] https://huggingface.co/huawei-noah/TinyBERT_General_4L_312D

Method	SST-2	MNLI-(m/mm)
MGSKD$_{\mathrm{m=1}}$	92.5	83.6/82.9
MGSKD$_{\mathrm{m=4}}$	92.9	83.9/83.3
MGSKD$_{\mathrm{m=16}}$	93.3	84.3/83.9
MGSKD$_{\mathrm{m=64}}$	93.7	84.7/84.3
MGSKD$_{\mathrm{m=128}}$	93.5	84.8/84.2

Table 2: The impact of relation heads.

rate as 1e-5, and the batch size as 32. We release our code to facilitate future research.[5]

4.3 Comparison Methods

Medium-sized Student Models. Most of the existing knowledge distillation methods are conducted on medium-sized student models which have 6 transformer layers, 768 hidden neurons, 12 attention heads, and overall 66M parameters. We adopt 3 of them as baselines: DistilBERT (Sanh et al., 2019), MiniLMv2 (Wang et al., 2021) and CKD (Park et al., 2021). Notice that these models adopted different distillation settings. DistilBERT and MiniLMv2 were firstly under task-agnostic distillation then directly fine-tuned on GLUE, while CKD was under both task-agnostic and task-specific distillation. The corpora they adopted for task-agnostic distillation were also not exactly the same. Nevertheless, we list the results as they reported on GLUE dev set as baselines, and we implement MiniLMv2 and CKD, two state-of-the-art distillation methods under the same distillation setting as ours for a fair comparison, which is described in the next paragraph.

Small-sized Student Models. For fair comparisons, we implement two state-of-the-art distillation methods: MiniLMv2 (Wang et al., 2021), CKD (Park et al., 2021) under the same distillation setting as ours. All these methods use the same student model as ours which has 4 transformer layers, 312 hidden neurons, 12 attention heads and overall 14M parameters. We adopt the fine-tuned ELECTRA$_{\mathrm{base}}$ as the teacher, and conduct task-specific distillation using the same distillation schedule and hyperparameters on the same augmented training sets as ours.

4.4 Main Results

We first evaluate the effectiveness of our proposed distillation framework. The main results are shown in Table 1. We calculate #Params by summing up the number of parameters contained in the embedding layer and all the transformer layers. The speed-up ratios are directly taken from previous works (Jiao et al., 2020; Wang et al., 2021). It can be observed that under the same distillation setting (models with † in Table 1), Student$^{\dagger}_{\mathrm{MGSKD}}$ outperforms strong baseline methods (i.e., Student$^{\dagger}_{\mathrm{MiniLMv2}}$ and Student$^{\dagger}_{\mathrm{CKD}}$) on 7 of the 8 GLUE tasks. When compared with medium-sized models from the literature which have more parameters but under different distillation settings (e.g., CKD), our method can still beat them on the majority of the 8 tasks. And surprisingly, with a stronger teacher model and data augmentation technique, our method MGSKD enables a 14M student transformer model to achieve comparable performance with BERT$_{\mathrm{base}}$ on most of the GLUE tasks, while keeping 9.4 times faster. Also, we observe that although MGSKD performs well on most of the GLUE tasks, it lags behind some baselines on CoLA, where the model is asked to judge the grammatical acceptability of a sentence. One reason might be that CoLA requires the model to focus on syntactic information while paying less attention to the sample-level semantic meanings, thus reducing the need for multi-granularity semantic knowledge that we propose to transfer to the student.

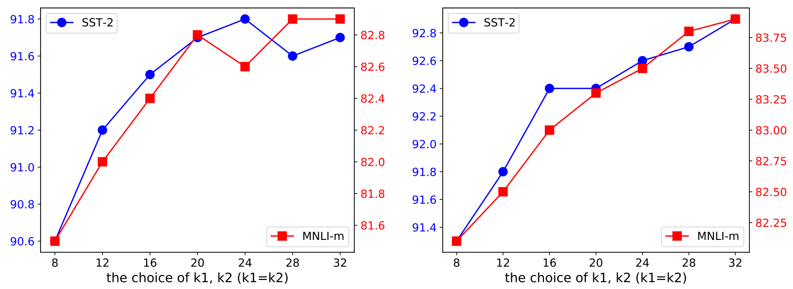
Method	SST-2	MNLI-(m/mm)
MGSKD	93.7	84.7/84.3
MGSKD w/o token	93.0	84.1/83.7
MGSKD w/o span	93.2	84.3/84.0
MGSKD w/o sample	92.8	83.9/83.6
MGSKD w token$_{\mathrm{p}}$	92.1	83.4/82.9
MGSKD w token$_{\mathrm{t}}$	91.7	82.8/82.6
MGSKD w token$_{\mathrm{p,t}}$	92.5	83.7/83.2
MGSKD w span$_{\mathrm{p}}$	91.8	82.5/82.3
MGSKD w span$_{\mathrm{t}}$	91.8	82.3/82.0
MGSKD w span$_{\mathrm{p,t}}$	92.2	83.0/82.7
MGSKD w sample$_{\mathrm{p}}$	91.9	82.6/82.5
MGSKD w sample$_{\mathrm{t}}$	92.9	83.9/83.5
MGSKD w sample$_{\mathrm{p,t}}$	92.8	83.7/83.6

Table 3: Ablation study of knowledge granularity. The subscripts $_{\mathrm{p}}$ and $_{\mathrm{t}}$ denote pair-wise and triplet-wise relations respectively.

4.5 Discussions

The Impact of Relation Heads. Recall that when calculating the structural relations between representations, we project them into m relation heads. We show how the number of relation heads impacts the performance on SST-2 and MNLI. As shown in Table 2, the performance gets better as the

[5] https://github.com/LC97-pku/MGSKD



(a) Token-level

(b) Sample-level

Figure 2: The accuracy curve of different k_1, k_2 for calculating angles.

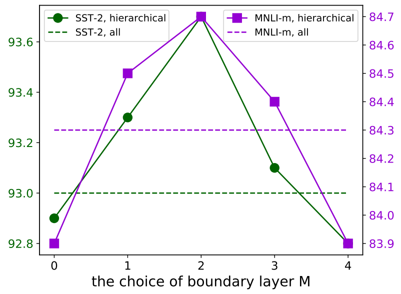


Figure 3: The accuracy curve of different choices of the boundary layer M.

number of relation heads increases, since it eases the trouble for the student to learn the structural relations in the very high-dimensional vector space by providing fine-grained supervision in multiple relatively low-dimensional spaces. We also find that when m is large, continuing to increase m is not worthwhile since the time and memory complexity increase linearly with m. Therefore we choose $m = 64$ in our setting.

Ablation Study of Knowledge Granularity. We transfer the structural knowledge to the student in three granularities: token-level, span-level, and sample-level. We extract pair-wise and triplet-wise structural relations for token- and span-level, while we adopt triplet-wise relations for sample-level. To verify the effectiveness of each granularity of knowledge and each form of structural relations, we conduct ablation studies and present the results in Table 3. (1) We first remove each granularity of knowledge from the objectives of MGSKD individually.[6] We can conclude that the sample-level knowledge is most crucial for the overall performance, the token-level knowledge provides moderate benefit, and the span-level knowledge contributes the least. We assume the reason why span-level knowledge distillation performs a little bit worse than token-level lies in that the average number of meaningful spans per sample on the 8 tasks is 7.19, which is 5.2 times fewer than the average number of tokens. Nevertheless, distillation with span-level knowledge still yields comparable performance. Overall, the results prove that each granularity of knowledge brings a positive effect to the model performance. (2) Then for each granularity, we study the effect of each form of structural knowledge (i.e., pair-wise and triplet-wise relations). In this stage, we distill each granularity of knowledge into all the student layers for a fair comparison. It can be observed that for token-level and span-level knowledge, pair-wise relations are more effective than triplet-wise relations, and the model performs better when jointly utilizing both. While for sample-level knowledge, we find that using triplet-wise relations outperforms using pair-wise relations by a large margin. Moreover, jointly utilizing the sample-level pair-wise and triplet-wise relations can't further improve the model's performance, therefore we only employ triplet-wise relations as sample-level knowledge.

The Impact of k_1 and k_2 for Calculating Angles. To ease the computation and memory complexity, we propose to sequentially select important representations to form angles, leading to the hyperparameters k_1 and k_2. We test different choices of k_1 and k_2 by adopting token-level and sample-level triplet-wise relations to teach the student respectively. To reduce the search space, we simply set $k_1 = k_2$. We draw the accuracy curve for different choices of k_1, k_2, as shown in Fig. 2. For token-level objectives, we find that increasing k_1, k_2 improves the accuracy when they are small and when $k_1, k_2 \geq 20$, the curves begin to vibrate. Therefore we choose $k_1 = k_2 = 20$ for token-level angle calculation. While for the triplet-wise relations of sample-level features, we observe that the accuracy increases monotonically with k_1, k_2. Therefore we just set k_1, k_2 as the batch size.

The Choice of the Boundary Layer M. We propose the hierarchical distillation strategy where we distill the token- and span-level knowledge into the bottom-M layers of the student and transfer the sample-level knowledge to the upper layers. To verify the effectiveness as well as to find the best choice of the boundary layer M, we conduct exper-

[6] When the sample-level objective is removed, we use the remaining objectives for all the student layers instead of only the bottom layers, as this setting yields better performance.

iments and show the results in Fig. 3. The dashed lines represent the setting dubbed as "*all*", where we distill token-, span- and sample-level knowledge into all the student layers. And the solid lines denote our hierarchical distillation setting with different choices of the boundary layer M. When $M = 0$ and $M = 4$, the student learns sample-level knowledge or token- and span-level knowledge for all layers. Without the help of other knowledge granularities, the student yields relatively poor performance on both tasks. As M increases from 0 to 4, we find the model's performance curves surpass the dashed lines, which verifies the effectiveness of our proposed hierarchical distillation strategy which transfers the knowledge to the proper positions of the student. We find the model achieves the highest accuracy when $M = 2$, i.e., the middle layer, indicating that both the syntactic knowledge transferred by token- and span-level features and the semantic knowledge derived from sample-level features are indispensable.

5 Conclusion

In this paper, we propose a novel knowledge distillation framework named MGSKD. We leverage intermediate representations of multi-granularity language units (i.e., tokens, spans and samples), and form the knowledge as the sophisticated structural relations between the representations rather than the individual representations themselves. The well-organized structural knowledge is then distilled into the student hierarchically across layers. Evaluation results on GLUE benchmark verify the effectiveness of our method. In the future, we plan to explore more forms of structural knowledge.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Key Research and Development Program of China (No. 2020AAA0106600).

Ethical Statement

This paper proposes a knowledge distillation framework that leverages multi-granularity structural knowledge to compress a large and powerful language model into a small one with minimum performance degradation, which is beneficial to energy-efficient NLP applications. The research will not pose ethical problems or negative social consequences. The datasets used in this paper are all publicly available and are widely adopted by researchers as the general testbed for natural language understanding evaluation. The proposed method doesn't introduce ethical/social bias or aggravate the potential bias in the data.

References

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2021. BinaryBERT: Pushing the limit of BERT quantization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4334–4348, Online. Association for Computational Linguistics.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs. *URL https://www. kaggle. com/c/quora-question-pairs*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2018. Universal transformers. *arXiv preprint arXiv:1807.03819*.

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

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*.

Hao Fu, Shaojun Zhou, Qihong Yang, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2021. Lrc-bert: Latent-representation contrastive knowledge distillation for natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12830–12838.

Mitchell Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing BERT: Studying the effects of weight pruning on transfer learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 143–155, Online. Association for Computational Linguistics.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *International Conference on Machine Learning*, pages 3690–3699. PMLR.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33:9782–9793.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2020. Mixkd: Towards efficient distillation of large-scale language models. *arXiv preprint arXiv:2011.00593*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Geondo Park, Gyeongman Kim, and Eunho Yang. 2021. Distilling linguistic context for language model compression. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 364–378, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976.

Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. 2019. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5007–5016.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Baitan Shao and Ying Chen. 2021. Multi-granularity for knowledge distillation. *Image and Vision Computing*, 115:104286.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8815–8821.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China. Association for Computational Linguistics.

Siqi Sun, Zhe Gan, Yuwei Fang, Yu Cheng, Shuohang Wang, and Jingjing Liu. 2020. Contrastive distillation on intermediate representations for language model compression. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 498–508, Online. Association for Computational Linguistics.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Frederick Tung and Greg Mori. 2019. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1365–1374.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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

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

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*, pages 36–39. IEEE.

Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020. TernaryBERT: Distillation-aware ultra-low bit BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 509–521, Online. Association for Computational Linguistics.