General Cross-Architecture Distillation of Pretrained Language Models into Matrix Embeddings — Supplementary Material —

Anonymized

Appendix

A. Overview of Architectural Choices

Figure 1 provides an overview of the architectural choices explored in this paper. We use pretrained BERT [1] as well as the embeddings from Mai et al. [2] as teacher for general distillation. Additionally, we pretrain a model CMOW/CBOW-Hybrid with our extension of masked language model training and bidirection on the same English Wikipedia + Toronto Books dataset. These pretrained embeddings may serve as initialization for the downstream classification models. We also evaluate downstream classification models that have been initialized randomly. For the downstream classification models, we consider three types of embeddings along with four types of classifiers.

For training on the downstream task, we use once again BERT as a teacher for task-specific distillation, while our experiments on general distillation only benefit from the initalization of the MLM-pretrained CMOW/CBOW-Hybrid model. Throughout the main part of the paper, we have reported scores with an MLP downstream classifier, which achieved the highest average scores. We have further experimented with a linear downstream classifier, LSTM, and 2D-CNN, which we briefly describe below.

a) LSTM: We have further experimented with pooling the sequence of embeddings with an LSTM. In the past, BiLSTM models have been successfully used in sentiment analysis tasks [3], [4]. In an LSTM network, the information at hand is propagated in the forward direction. Thus, each state t depends on its predecessor $t-1$. BiLSTM are LSTM networks, in which the inputs are processed twice: once in the forward direction and once in the backward direction, generating a set of two outputs. In order to generate the output vectors, the output of a single BiLSTM block is fed into an MLP, consisting of two consecutive linear layers with ReLU activation functions. Note that the BiLSTM operates on a sequence of token embeddings, instead of operating on pooled sentence embeddings like the other student models. We apply a dropout of 0.5 after the first linear layer.

b) CNN: We also explore a 2D-CNN classifier that induces a bias for learning two-dimensional structures within the (aggregated) embedding matrices. The CNN consists of one transposed convolution, which increases the matrix dimensions by a factor of four. Following that, we employ a block of three convolutional layers, the first one having a single filter (or two, for hybrid variants) and a kernel size of four, with the remaining two layers having 3 (4) kernels with stride 2. To avoid distorting the input embeddings, no padding is applied. ReLU is used for all activation functions. We apply BatchNorm for regularization before the last convolutional layer's output is flattened and passed into a linear layer, which produces the predictions. We add a dropout of 0.4 before the last linear layer.

B. Discussion of Hyperparameters and Loss Functions for Distillation

We list hyperparameter search spaces along with their optimization methods in Table I. For the experiments on data augmentation and using only soft loss, we keep the configurations of the best models (See Table II and tune the learning rate, again. We optimize over all six initial learning rates, namely $\{10^{-3}$, $5 \cdot 10^{-4}$, 10^{-4}, $5 \cdot 10^{-5}$, and $10^{-5}\}$. All initial learning rates decay linearly over the course of training. Note, we also experimented with using warmup steps versus no warmup for the learning rate schedule. As the warmup did not improve the results, we did not use it.

For the softmax temperature, we find that $T = 1$ is often used [5], [6], [7], [8], [9]. Since a higher temperature also flattens the curve over all predictions, it could add too much noise and it is therefore better to use a smaller temperature [10]. Setting the weight $\alpha = 1$ corresponds to only using hard loss and $\alpha = 0$ to only using soft loss. Since we do not want to discard any information stemming from the hard loss, we do not follow the approach of Wasserblat et al. who only use the soft loss [11] but instead, we employ a vanilla knowledge distillation approach following Hinton et al. [5].

Hinton et al. [5] state, that using cross-entropy loss on the softmax-temperature with a large temperature, for example $T = 20$, corresponds to only using the Mean Square Error (MSE) loss on the raw student and teacher logits. Therefore it is also common to use this loss for the soft distillation loss [12], [11], [13]. While Tang et al. [12] used the weighted hard cross-entropy loss in the overall loss calculation, Wasserblat et al. and Mukherjee et al. only used the soft loss [12], [11], [13]. A disadvantage of MSE loss is that every error has a huge effect on the overall loss, since it is squared. Another point is, that Hinton et al. found it beneficial to use small temperature values if the teacher is way bigger than the student [5]. Since using MSE loss corresponds to using big T values, this loss does not apply to our use case of using small students for lower

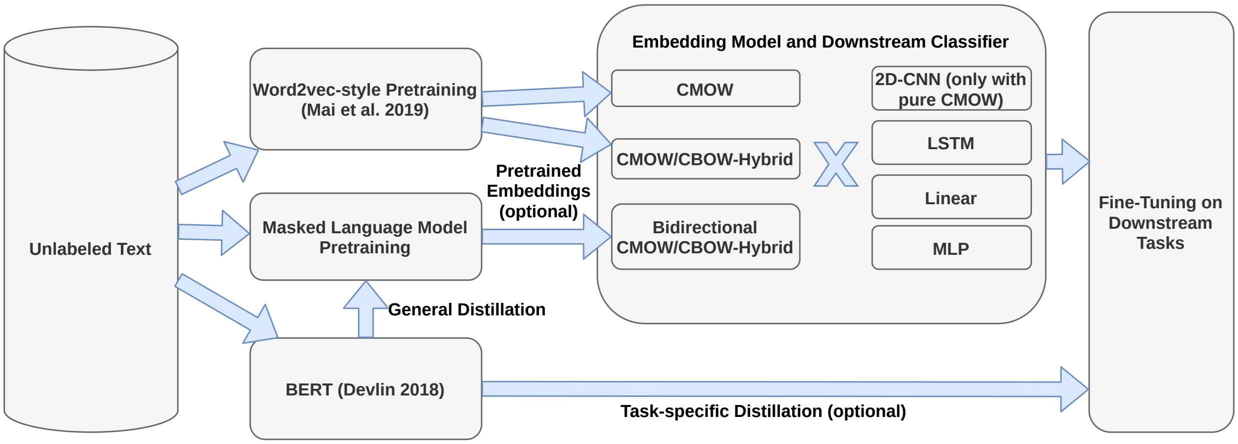
Unlabeled Text
Word2vec-style Pretraining (Mai et al. 2019)
Masked Language Model Pretraining
BERT (Devlin 2018)
General Distillation
Pretrained Embeddings (optional)
Embedding Model and Downstream Classifier
CMOW
CMOW/CBOW-Hybrid
Bidirectional CMOW/CBOW-Hybrid
2D-CNN (only with pure CMOW)
LSTM
Linear
MLP
Fine-Tuning on Downstream Tasks
Task-specific Distillation (optional)

Fig. 1. All considered for embeddings and downstream classifiers, pretraining and fine-tuning.

TABLE I
HYPERPARAMETER SEARCH SPACE AND OPTIMIZATION METHOD

Hyperparameter	Range	Opt. method
	— General Distillation —	
Learning rate	$\{10^{-3}, 5\cdot10^{-4}, 10^{-4}, 5\cdot10^{-5}, 10^{-5}\}$	grid search
Warmup steps	$\{0, 500\}$	grid search
Embedding dropout	$\{0, 0.1\}$	grid search
Hidden unit dropout	$\{0.2\}$	fixed
Batch size	{1,8,32,64,128,256}	manual
	— Task-specific Distillation —	
Learning rate	$\{10^{-3}, 5\cdot10^{-4}, 10^{-4}, 5\cdot10^{-5}, 10^{-5}, 5\cdot10^{-6}\}$	grid search
Embedding type	Hybrid, CMOW, CBOW	grid search
Embedding initialization	random, pretrained	grid search
DiffCat	true, false	grid search
Bidirectional	true, false	grid search
Classifier	Linear Probe, MLP, CNN, BiLSTM	grid search

bound knowledge distillation, but with cross-entropy loss, we still have the possibility to achieve the behavior of the MSE loss by setting the value of T to a big value.

C. Detailed Results

In the following, we provide detailed results for task-specific distillation including the different downstream classifiers, unidirectional CMOW/CBOW-Hybrid, and joint two-sequence encoding. The best performing model per task are marked in bold. We abbreviate CMOW/CBOW-Hybrid as 'Hybrid'.

For the unidirectional baseline model CMOW/CBOW-Hybrid, we initialize with pretrained embeddings provided by Mai et al. [2][1], which cover 54% of BERT's vocabulary. As initialization for the newly developed bidirectional CMOW/CBOW-Hybrid models, we use our own pretrained model obtained by general distillation with BERT.

Table II summarizes the best performing models along with their hyperparameter configurations for each task. Note that we have chosen to use an MLP downstream classifier for the results reported in the main part of this work. Using an MLP downstream classifier has led to the highest average scores across all GLUE tasks.

In Table III, we report an extended version of the comparison with the literature. Here, we also include our BERT-base teacher model, aswell as TinyBERT [7] and Tang et al. [12]'s distilled BiLSTM. Note that TinyBERT and BiLSTM are not fully comparable, because those numbers are reported on the official GLUE test set, while we have used the validation set for our experiments.

In Table IV, we report the results for all downstream classifiers without the DiffCat aggregation but with a sequential BERT-like two-sequence encoding. It is interesting to see that the CMOW-only variant with 2D-CNN classifier leads to the best scores on sentiment analysis task SST-2. Note that all CMOW variants reported in this table are unidirectional and use task-specific distillation.

In Table V, we report the results for all downstream classifiers with DiffCat two-sequence encoding. Here we observe, that pretrained CBOW with an MLP classifier leads to the best results on sentence similarity (STS-B). Again, all CMOW variants reported in this table are unidirectional and use task-specific distillation.

[1] Downloaded from Zenodo: https://zenodo.org/record/3933322#.YKJ_uxKxXJU

In Table VI, we report the results for bidirectional models with DiffCat two-sequence encoding.

From all tables combined, we see that Bidirectional CMOW/CBOW-Hybrid model leads to the highest scores on average, even though, on individual tasks, some other variations of the approach lead to higher scores. Thus, we regard bidrectional CMOW/CBOW-Hybrid as our primary model, whose scores we have reported in the main paper, while isolating the effect of the individual components (bidirection, DiffCat encoding, distillation strategies).

We list the number of parameters in Tables VII and VIII. While the absolute numbers might seem high, it is important to note that we have also counted the parameters of the embeddings. As we show in the tables, the number of parameters in the classification models is much lower.

We have performed further experiments with the best performing model for each task: data augmentation and using only soft loss.

a) Using Only Soft Loss: We study the influence of the alpha value used in the loss function, based on the best results obtained with the initial $\alpha = 0.5$. The goal is to investigate whether using only soft loss, i. e., setting $\alpha = 0.0$ leads to different results. As Table III shows, using only soft loss improves only the MRPC task by a small margin.

b) Data Augmentation: We conduct a further study with data augmentation as in TinyBERT [7]. We employ their technique of replacing words by similar word embeddings and nearest predictions from BERT to augment the GLUE training datasets. The results are shown in Table III. We find that the effect of data augmentation is small. An improvement was only observed on SST-2 (+1.2 points) and STS-B (+3.6).

References

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT (1)*. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[2] F. Mai, L. Galke, and A. Scherp, "CBOW is not all you need: Combining CBOW with the compositional matrix space model," in *ICLR (Poster)*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=H1MgjoR9tQ

[3] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on bilstm," *IEEE Access*, vol. 7, pp. 51 522–51 532, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2909919

[4] Z. Hameed and B. Garcia-Zapirain, "Sentiment classification using a single-layered bilstm model," *IEEE Access*, vol. 8, pp. 73 992–74 001, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2988550

[5] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv*, vol. abs/1503.02531, 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[6] Y. Mao, Y. Wang, C. Wu, C. Zhang, Y. Wang, Q. Zhang, Y. Yang, Y. Tong, and J. Bai, "Ladabert: Lightweight adaptation of BERT through hybrid model compression," in *COLING*. International Committee on Computational Linguistics, 2020, pp. 3225–3234. [Online]. Available: https://doi.org/10.18653/v1/2020.coling-main.287

[7] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," in *EMNLP (Findings)*. Association for Computational Linguistics, 2020, pp. 4163–4174. [Online]. Available: https://doi.org/10.18653/v1/2020.findings-emnlp.372

[8] A. K. Mishra and D. Marr, "Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy," in *ICLR (Poster)*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=B1ae1lZRb

[9] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," in *ICLR (Poster)*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=S1XolQbRW

[10] G. Chen, W. Choi, X. Yu, T. X. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *NIPS*, 2017, pp. 742–751. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/e1e32e235eee1f970470a3a6658dfdd5-Abstract.html

[11] M. Wasserblat, O. Pereg, and P. Izsak, "Exploring the boundaries of low-resource BERT distillation," in *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*. Association for Computational Linguistics, Nov. 2020, pp. 35–40. [Online]. Available: https://aclanthology.org/2020.sustainlp-1.5

[12] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova, and J. Lin, "Distilling task-specific knowledge from bert into simple neural networks," *ArXiv*, vol. abs/1903.12136, 2019. [Online]. Available: https://arxiv.org/abs/1903.12136

[13] S. Mukherjee and A. H. Awadallah, "Distilling bert into simple neural networks with unlabeled transfer data," *ArXiv*, vol. abs/1910.01769, 2020. [Online]. Available: https://arxiv.org/abs/1910.01769

[14] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2020. [Online]. Available: https://arxiv.org/abs/1910.01108

[15] G. Phua, S. Lin, and D. Poletti, "Word2rate: training and evaluating multiple word embeddings as statistical transitions," *ArXiv*, vol. abs/2104.08173, 2021. [Online]. Available: https://arxiv.org/abs/2104.08173

[16] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "MobileBERT: a compact task-agnostic BERT for resource-limited devices," in *ACL*. Association for Computational Linguistics, 2020, pp. 2158–2170. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.195

TABLE II
Hyperparameter configurations for best-performing models by GLUE task

Task	**Score**	**Classifier**	**Emb. type**	**Emb. initialization**	**DiffCat**	**Bidirectional**	**Learning rate**
CoLA	23.3	MLP	CMOW/CBOW-Hybrid	pretrained	true	true	1.0E-4
MNLI-m	63.3	MLP	CMOW/CBOW-Hybrid	not pretrained	true	true	1.0E-4
MRPC	78.2	MLP	CBOW	pretrained	true	false	1.0E-3
QNLI	72.6	MLP	CMOW/CBOW-Hybrid	not pretrained	true	true	5.0E-5
QQP	86.6	MLP	CMOW/CBOW-Hybrid	not pretrained	true	false	1.0E-4
RTE	59.9	MLP	CMOW/CBOW-Hybrid	pretrained	true	true	5.0E-4
SST-2	86.8	CNN	CMOW	not pretrained	false	false	5.0E-4
STS-B	66.0	MLP	CBOW	pretrained	true	false	1.0E-4
WNLI	69.0	CNN	CMOW	pretrained	false	false	1.0E-5

TABLE III
Scores on the GLUE development set. Our best performing general distillation and task-specific distillation models are highlighted in bold font per task. References indicate sources of scores. The $\star$-symbol indicates numbers on the official GLUE test set. CMOW/CBOW-Hybrid is abbreviated as 'Hybrid'.

	Score	CoLA	MNLI-m	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
— large-scale pre-trained language models —										
ELMo [14]	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base [14]	79.5	56.3	86.7	88.6	91.7	89.6	69.3	92.7	89.0	53.5
BERT-base (our teacher model)	78.9	57.9	84.2	84.6	91.4	89.7	67.9	91.7	88.0	54.9
Word2rate [15]	—	—	—	—	—	—	—	65.7	53.1	—
— general distillation baselines —										
DistilBERT [14]	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3
$\star$ MobileBERT [16]	—	51.1	84.3	88.8	91.6	70.5	70.4	92.6	84.8	—
— task-specific distillation baselines —										
$\star$ TinyBERT (4 layers) [7]	—	44.1	82.5	86.4	87.7	71.3	66.6	92.6	80.4	—
CBOW-FFN [11]	—	10.0	—	—	—	—	—	79.1	—	—
BiLSTM [11]	—	10.0	—	—	—	—	—	80.7	—	—
— general distillation (ours) —										
Bidi. Hybrid + Linear	65.1	15.0	63.6	**80.9**	70.7	84.3	56.7	**84.0**	71.1	**59.2**
Bidi. Hybrid + MLP	66.6	**16.7**	**66.6**	79.7	**71.7**	**87.2**	**61.0**	82.9	**76.9**	56.3
— task-specific distillation (ours) —										
CMOW + CNN (rand. init.)	54.6	13.4	45.6	72.3	61.2	82.6	56.3	**86.8**	15.0	57.8
CMOW + CNN (pretrained)	56.2	18.3	50.1	71.8	60.5	80.6	57.0	85.0	13.2	**69.0**
CBOW + MLP (pretrained)	63.8	14.0	61.7	**78.2**	70.8	86.2	57.4	83.8	**66.0**	56.3
Hybrid + MLP (rand. init.)	62.5	13.1	62.5	74.3	71.5	**86.6**	58.1	83.1	58.6	56.3
Bidi. Hybrid + MLP (rand. init.)	63.2	13.0	**63.3**	75.7	**72.6**	86.1	57.4	83.3	59.7	57.7
Bidi. Hybrid + MLP (pretrained)	64.6	**23.3**	61.8	75.0	72.0	86.3	**59.9**	82.9	62.9	57.7
— further experiments on best-performing task-specific distillation models —										
Only soft loss ($\alpha = 0$)	64.0	19.9	62.3	78.7	72.4	68.5	56.3	86.6	62.4	69.0
Data augmentation	63.5	21.2	47.3	76.2	72.1	86.6	52.7	88.0	69.6	57.7

TABLE IV

Scores on the GLUE development set without DiffCat encoding

Task-Specific Distillation	Score	CoLA	MNLI-m	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
— task-specific finetuning (ours) —										
Teacher BERT-base	78.9	57.9	84.2	84.6	91.4	89.7	67.9	91.7	88.0	54.9
— task-specific distillation (ours) CBOW not pretrained —										
Linear probe	52.8	12.2	43.0	72.3	60.1	74.8	55.6	82.8	17.7	56.3
MLP	53.2	13.0	46.3	71.3	59.7	76.9	54.5	82.9	17.5	56.3
CNN	52.8	11.7	43.0	72.1	60.1	77.5	54.5	82.7	17.2	56.3
BiLSTM	52.1	10.9	44.9	70.8	59.8	78.1	54.5	81.3	12.3	56.3
— task-specific distillation (ours) CBOW pretrained —										
Linear probe	52.4	11.0	43.2	72.1	58.8	74.8	54.9	82.5	14.0	60.6
MLP	54.0	14.3	46.3	71.3	60.1	76.9	58.5	83.1	14.8	60.6
CNN	53.0	12.0	43.5	71.6	59.2	77.5	55.2	82.6	18.8	56.3
BiLSTM	50.8	0	44.9	71.3	59.4	78.0	54.0	81.0	12.0	56.3
— task-specific distillation (ours) CMOW not pretrained —										
Linear probe	53.7	13.8	45.3	72.1	62.5	80.9	53.4	84.1	15.2	56.3
MLP	54.8	15.1	45.6	72.8	60.6	82.6	55.6	84.3	20.0	56.3
CNN	54.6	13.4	45.6	72.3	61.2	82.6	56.3	**86.8**	15.0	57.8
BiLSTM	53.2	16.7	44.9	72.1	64.8	80.6	54.2	82.9	7.9	54.9
— task-specific distillation (ours) CMOW pretrained —										
Linear probe	54.3	20.8	48.6	71.3	60.3	78.4	54.9	84.5	13.8	56.3
MLP	55.4	18.9	50.4	72.3	61.3	79.3	55.2	83.0	17.9	60.6
CNN	56.2	18.3	50.1	71.8	60.5	80.6	57.0	85.0	13.2	**69.0**
BiLSTM	51.4	0	44.2	68.4	59.8	81.1	55.2	82.3	15.0	56.3
— task-specific distillation (ours) CMOW/CBOW-Hybrid not pretrained —										
Linear probe	54.4	17.0	47.0	72.6	61.1	81.4	53.4	84.5	15.1	57.8
MLP	54.4	13.8	50.0	73.0	60.4	78.6	53.8	84.9	18.5	56.3
CNN	53.6	12.0	42.1	72.6	60.9	79.6	52.7	85.7	16.3	60.6
BiLSTM	52.4	0	43.2	72.1	61.2	80.0	57.4	83.0	18.1	56.3
— task-specific distillation (ours) CMOW/CBOW-Hybrid pretrained —										
Linear probe	53.9	19.1	41.0	71.8	57.6	78.7	57.8	83.7	16.2	59.2
MLP	55.3	22.1	47.4	71.6	60.0	79.5	57.8	84.1	18.1	56.3
CNN	54.0	20.7	44.5	71.8	59.9	79.7	54.9	85.9	9.9	59.1
BiLSTM	53.7	17.0	40.6	71.8	61.3	80.3	57.4	82.5	14.0	59.2

TABLE V
SCORES ON THE GLUE DEVELOPMENT SET WITH DIFFCAT TWO-SEQUENCE ENCODING

Task-Specific Distillation	Score	CoLA	MNLI-m	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
— task-specific finetuning (ours) —										
Teacher BERT-base	78.9	57.9	84.2	84.6	91.4	89.7	67.9	91.7	88.0	54.9
— task-specific distillation (ours) CBOW not pretrained —										
Linear probe	53.8	11.5	46.6	72.8	62.2	76.7	52.7	83.5	22.0	56.3
MLP	61.0	14.3	57.8	77.2	70.3	86.0	56.7	82.3	47.0	**57.7**
CNN	53.8	11.2	51.5	75.0	65.8	81.3	53.1	82.3	7.2	56.3
BiLSTM	48.4	11.5	31.8	68.3	66.8	63.2	56.7	83.5	1.5	56.3
— task-specific distillation (ours) CBOW pretrained —										
Linear probe	56.3	9.0	47.1	72.8	64.8	77.1	53.4	82.5	43.4	56.3
MLP	63.8	14.0	61.7	**78.2**	70.8	86.2	57.4	83.8	**66.0**	56.3
CNN	53.7	10.9	55.0	73.8	66.2	82.1	53.1	82.2	3.8	56.3
BiLSTM	47.7	0	32.7	68.4	69.6	63.2	55.6	82.5	1.3	56.3
— task-specific distillation (ours) CMOW not pretrained —										
Linear probe	55.1	10.9	54.3	71.8	62.7	80.9	56.0	85.2	17.6	56.3
MLP	63.2	14.2	61.9	75.5	72.4	86.3	55.2	83.7	62.7	56.3
CNN	55.4	12.4	45.3	72.3	61.5	82.6	57.4	84.3	26.1	56.3
BiLSTM	47.5	0	31.8	70.3	49.5	81.0	55.6	83.4	0	56.3
— task-specific distillation (ours) CMOW pretrained —										
Linear probe	56.3	22.4	48.4	72.5	61.3	81.9	54.5	83.9	24.2	**57.7**
MLP	61.2	20.9	60.2	73.8	64.6	85.9	54.9	84.4	49.4	56.3
CNN	53.4	18.5	40.6	71.8	58.2	68.3	54.9	85.4	26.9	56.3
BiLSTM	49.7	0	32.7	68.3	67.2	82.9	57.0	82.5	0	56.3
— task-specific distillation (ours) Hybrid not pretrained —										
Linear probe	51.7	11.2	39.0	71.1	49.5	81.8	56.0	85.2	14.3	**57.7**
MLP	62.5	13.1	62.5	74.3	71.5	**86.6**	58.1	83.1	58.6	56.3
CNN	52.8	11.9	45.3	71.6	61.4	84.8	55.2	85.4	2.9	56.3
BiLSTM	50.9	0	42.6	70.1	60.3	79.3	56.0	84.4	9.3	56.3
— task-specific distillation (ours) Hybrid pretrained —										
Linear probe	54.0	19.6	45.7	71.3	63.4	80.9	54.2	84.1	11.7	54.9
MLP	62.7	20.9	62.6	74.5	68.6	85.7	56.3	83.1	56.2	56.3
CNN	57.9	19.6	37.6	75.7	62.0	85.4	54.9	82.3	48.5	54.9
BiLSTM	52.1	0	48.0	68.4	71.9	85.3	56.7	82.5	0	56.3

TABLE VI
SCORES ON THE GLUE DEVELOPMENT SET WITH DIFFCAT ENCODING AND THE BIDIRECTIONAL CMOW/CBOW-HYBRID MODEL

Task-Specific Distillation	Score	CoLA	MNLI-m	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
— task-specific finetuning (ours) —										
Teacher BERT-base	78.9	57.9	84.2	84.6	91.4	89.7	67.9	91.7	88.0	54.9
— task-specific distillation (ours) Bidirectional Hybrid, not pretrained —										
Linear probe	53.5	11.6	39.4	71.6	64.3	82.5	56.3	85.0	14.6	56.3
MLP	63.2	13.0	**63.3**	75.7	**72.6**	86.1	57.4	83.3	59.7	57.7
CNN	52.7	14.5	37.3	71.3	60.8	86.4	55.2	85.8	6.6	56.3
— task-specific distillation (ours) Bidirectional Hybrid, pretrained —										
Linear probe	55.5	18.1	42.4	72.1	64.9	81.2	56.7	85.2	22.5	56.3
MLP	**64.6**	**23.3**	61.8	75.0	72.0	86.3	**59.9**	82.9	62.9	57.7
CNN	55.1	20.5	39.3	73.8	61.3	85.9	56.3	85.5	15.9	57.7

TABLE VII
Number of parameters without DiffCat encoding

	CoLA, MRPC, QNLI, QQP, SST-2, RTE, WNLI	MNLI	STS-B
	— task-specific distillation (ours) CBOW —		
Linear probe	47,861,634	47,862,419	47,876,549
– only classifier	3,138	3,923	18,053
MLP	48,647,498	48,648,499	48,666,517
– only classifier	789,002	790,003	808,021
CNN	47,862,708	47,864,737	47,901,259
– only classifier	4,212	6,241	42,763
BiLSTM	53,704,002	53,705,027	53,723,477
– only classifier	5,845,506	5,846,531	5,864,981
	— task-specific distillation (ours) CMOW —		
Linear probe	23,932,386	23,933,171	23,947,301
– only classifier	3,138	3,923	18,053
MLP	24,718,250	24,719,251	24,737,269
– only classifier	789,002	790,003	808,021
CNN	23,933,460	23,935,489	23,972,011
– only classifier	4,212	6,241	42,763
BiLSTM	24,853,978	24,854,371	35,022,869
– only classifier	924,730	925,123	110,936,21
	— task-specific distillation (ours) Hybrid —		
Linear probe	24,420,802	24,421,603	24,436,021
– only classifier	3,202	4,003	18,421
MLP	25,222,602	25,223,603	25,241,621
– only classifier	805,002	806,003	824,021
CNN	24,420,558	24,421,855	24,445,201
– only classifier	2,958	4,255	27,601
BiLSTM	30,328,642	30,329,667	30,348,117
– only classifier	5,911,042	5,912,067	5,930,517

TABLE VIII
NUMBER OF PARAMETERS WITH DIFFCAT TWO-SEQUENCE ENCODING

	CoLA, MRPC, QNLI, QQP, SST-2, RTE, WNLI	MNLI	STS-B
— task-specific distillation (ours) CBOW —			
Linear probe	47,867,906	47,870,259	47,912,613
– only classifier	9,410	11,763	54,117
MLP	50,215,498	50,216,499	50,234,517
– only classifier	2,357,002	2,358,003	2,376,021
CNN	47,865,932	47,869,313	47,930,171
– only classifier	7,436	10,817	71,675
BiLSTM	147,477,458	147,479,811	147,522,165
– only classifier	99,618,962	99,621,315	99,663,669
— task-specific distillation (ours) CMOW —			
Linear probe	23,938,658	23,941,011	23,983,365
– only classifier	9,410	11,763	54,117
MLP	26,286,250	26,287,251	26,305,269
– only classifier	2,357,002	2,358,003	2,376,021
CNN	23,936,684	23,940,065	24,000,923
– only classifier	7,436	10,817	71,675
BiLSTM	123,548,210	123,550,563	123,592,917
– only classifier	99,618,962	99,621,315	99,663,669
— task-specific distillation (ours) Hybrid —			
Linear probe	24,427,202	24,429,603	24,472,821
– only classifier	9,602	12,003	55,221
MLP	26,822,602	26,823,603	26,841,621
– only classifier	2,405,002	2,406,003	2,424,021
CNN	24,424,990	24,427,583	24,474,257
– only classifier	7,390	9,983	56,657
BiLSTM	128,143,202	128,145,603	128,188,821
– only classifier	103,725,602	103,728,003	103,771,221
— task-specific distillation (ours) Hybrid bidirectional —			
Linear probe	36,640,802	36,644,403	36,709,221
– only classifier	14402	18003	82,821
MLP	40,231,402	40,232,403	4,025,0421
– only classifier	3,605,002	3,606,003	3,624,021
CNN	36,638,164	36,641,729	36,705,899
– only classifier	11,764	15,329	79,499
BiLSTM	451,437,602		451,528,821
– only classifier	414,811,202		414,902,421