Efficient Transformer Knowledge Distillation: A Performance Review

Nathan Brown[1], Ashton Williamson[1], Tahj Anderson[1], and Logan Lawrence[2,*]

[1]School of Computing, Clemson University, [2]Giant Oak Inc.
{nbrown9,taw2,tahja}@clemson.edu, logan.lawrence@giantoak.com

Abstract

As pretrained transformer language models continue to achieve state-of-the-art performance, the Natural Language Processing community has pushed for advances in model compression and efficient attention mechanisms to address high computational requirements and limited input sequence length. Despite these separate efforts, no investigation has been done into the intersection of these two fields. In this work, we provide an evaluation of model compression via knowledge distillation on efficient attention transformers. We provide cost-performance trade-offs for the compression of state-of-the-art efficient attention architectures and the gains made in performance in comparison to their full attention counterparts. Furthermore, we introduce a new long-context Named Entity Recognition dataset, GONERD, to train and test the performance of NER models on long sequences. We find that distilled efficient attention transformers can preserve a significant amount of original model performance, preserving up to **98.6%** across short-context tasks (GLUE, SQUAD, CoNLL-2003), up to **94.6%** across long-context Question-and-Answering tasks (HotpotQA, TriviaQA), and up to **98.8%** on long-context Named Entity Recognition (GONERD), while decreasing inference times by up to **57.8%**. We find that, for most models on most tasks, performing knowledge distillation is an effective method to yield high-performing efficient attention models with low costs.

1 Introduction

The rise of Transformer-based models (Vaswani et al., 2017) has driven significant advancements in the field of Natural Language Processing (NLP). Of these models, BERT (Devlin et al., 2018; Rogers et al., 2020) produced landmark performance in a variety of NLP tasks such as Question Answering (QA), Named Entity Recognition (NER), and

GLUE (Wang et al., 2018). BERT-based models (Rogers et al., 2020) continue to dominate the field (Zhou et al., 2023) with variations such as RoBERTa (Liu et al., 2019) dramatically improving performance on downstream tasks.

However, BERT-based models often have a fairly short maximum input length of 512 tokens, severely limiting their capabilities in long-context situations. Attempting to increase this limit to allow for longer sequences often results in significantly greater computational requirements. This has given rise to the creation of *efficient* attention transformer models (Tay et al., 2022) such as Longformer (Beltagy et al., 2020), Big Bird (Zaheer et al., 2020), Nyströmformer (Xiong et al., 2021), and LSG (Condevaux and Harispe, 2023), which can accept as input much longer sequences with reduced computational overhead by modifying and approximating BERT's original attention mechanism.

While efficient attention models require less computational resources on long-context tasks when compared to their non-efficient counterparts, they are still often computationally expensive to train and deploy (Sharir et al., 2020). Thus, organizations and individuals are required to grapple with increased operational costs, difficulty deploying these models on resource-limited hardware such as mobile devices, and often must rely on cloud-based solutions which restricts model availability in scenarios with limited internet access.

In response to computational challenges associated with transformer models, the NLP community has invested considerable efforts into creating cheaper yet performant models. This has been particularly the case in the study of Knowledge Distillation (KD) (Gou et al., 2021; Hinton et al., 2015). However, despite the rapid progress of KD and its effectiveness in model compression, little work has been done toward the investigation of the intersection of KD and efficient attention architectures. As

---

*Corresponding author.

such, we focus on combining these two methodologies. We believe this is an essential effort for creating models that can cheaply and effectively operate on a production scale on long-context tasks. Furthermore, despite its significance in practical NLP usage, Named Entity Recognition (NER) still does not have a well-accepted long-context benchmark. Our work attempts to address these two needs directly.

The main contributions of this work are twofold:

1. Performance analysis of popular pretrained efficient transformers and their distilled students in various contexts, including **GLUE**, **SQuAD** (Rajpurkar et al., 2016, 2018), **HotpotQA** (Yang et al., 2018), **TriviaQA** (Joshi et al., 2017), **CoNLL-2003** (Tjong Kim Sang and De Meulder, 2003), and **GONERD**.

2. The introduction of a new long-context NER task: the **G**iant **O**ak **NER D**ataset (**GONERD**). This dataset and all models are publicly available on Hugging Face[*].

In particular, we find that distilling Longformer-RoBERTa (Beltagy et al., 2020) yields the best results during our experiments, producing substantial improvements in cost performance over state-of-the-art models. In short, it retains considerable performance on GLUE (**92.3%**), SQuAD (**93.0%**), HotpotQA (**88.4%**), CoNLL-2003 (**99.8%**), and GONERD (**95.9%**) while decreasing inference times by **49.3%** on long sequences. In the context of GONERD, this is effectively **95.9%** of the original model's performance for **50.7%** of the cost.

2 Related Work

Considerable success has been made in the compression of BERT (Devlin et al., 2018) which, at the time of its release, was one of the largest models in NLP. BERT itself has been expanded to fit many different use cases including, but not limited to, RoBERTA (Liu et al., 2019), a model built to improve BERT performance on a variety of tasks through clever choices in training data and hyperparameters, XLM-R (Conneau et al., 2020), which was built using similar methods on extremely multilingual data (100 languages), and DistilBERT (Sanh et al., 2019), which sought to greatly reduce

---
[*]https://huggingface.co/giant-oak

the computational costs of BERT through knowledge distillation. ALBERT (Lan et al., 2019) factorizes the embedding matrices of BERT and shares weights between layers to significantly decrease the parameter size, thereby decreasing training and inference costs.

BERT-based distillation methods, such as DistilBERT (Sanh et al., 2019), TinyBERT (Jiao et al., 2020), and MobileBERT (Sun et al., 2020) have gained prominence due to their utilization of distillation techniques and can be applied to a wide variety of BERT-based architectures. These models have significantly reduced the computational requirements and resource consumption associated with BERT-based NLP models, making them more accessible and easily deployable on resource-constrained hardware. However, BERT's attention mechanism still results in a quadratic dependency on sequence length, resulting in greater computational requirements at higher sequence lengths.

To solve this problem with BERT-based architectures, methods have been developed to create *efficient* attention transformer models (Tay et al., 2022) which can operate on sequences many times longer than their BERT counterparts. Two popular methods in this area are Longformer (Beltagy et al., 2020) and Big Bird (Zaheer et al., 2020), which use dilated sliding window and a combination of global, sliding, and random activations in their attention matrices, respectively, to increase the maximum input sequence length from 512 to 4096. More recently, Local Sparse Global (LSG) (Condevaux and Harispe, 2023) attention uses a Locality Sensitive Hashing algorithm (Andoni et al., 2015) with the Local, Sparse, and Global patterns used in Longformer and Big Bird, whereas Nyströmformer (Xiong et al., 2021) uses a Nyström matrix approximation to the regular softmax attention, reducing self-attention complexity to linear time.

The **L**ong-**R**ange **A**rena (**LRA**) (Tay et al., 2021), a comprehensive suite of benchmarking tasks toward systematically evaluating long-context transformer architectures, is novel in that its tasks largely decouple the effect of Masked Language Modeling (MLM) pretraining from efficient model performance. While useful for developing new transformer architectures, we are primarily focused on the comparative performance between student and teacher models on downstream tasks after having been pretrained/distilled on MLM. As such,

LRA is not utilized in this paper.

3 Methodology

3.1 Knowledge Distillation

Knowledge Distillation (KD) for transformer-based architectures (Gou et al., 2021) is most commonly executed in three steps: (1) Pretrain a larger, complex model. (2) Distill knowledge from the larger complex model into a smaller, simpler model. (3) Fine-tune the student model on a downstream task. While effective in short-context scenarios, this three-step process leaves room for ambiguity regarding the recommended distillation process for long-context efficient attention transformer models.

In this paper, we use the term *"convert"* to refer to the process of updating a pretrained LM to use an efficient attention pattern, i.e. one capable of input lengths longer than 512 tokens. Considering this, we can identify two possible methods of inserting the conversion operation into the classical KD pipeline:

1. **Convert-Then-Distill**: Convert teacher → Pretrain teacher → Distill into student → Fine-tune student on downstream task

2. **Distill-Then-Convert**: Pretrain teacher → Distill into student → Convert student → Fine-tune student on downstream task

Although **Distill-Then-Convert** is conceptually interesting and potentially fruitful, we will be only covering **Convert-Then-Distill** in this work. However, we do include an experiment directly extending the maximum input sequence length in Section 4.2 of existing non-efficient distilled students, demonstrating the necessity of the *conversion* step in long-context tasks in terms of reducing a model's computational requirements.

Within the realm of **Convert-Then-Distill**, we perform knowledge distillation using the same process utilized in the creation of DistilBERT (Sanh et al., 2019). Namely, we begin by compressing pretrained efficient teacher models Longformer RoBERTa (Beltagy et al., 2020), Big Bird RoBERTa (Zaheer et al., 2020), LSG RoBERTa (Condevaux and Harispe, 2023), and Nyström-former (Xiong et al., 2021). In all cases, the number of hidden layers is reduced by a factor of two, with the student model being initialized by taking every other hidden layer from the teacher.

During training, the distillation loss is calculated over the soft target probabilities of the teacher. A softmax temperature is used, and a linear combination of the distillation loss, MLM supervised training loss, and cosine embedding loss is performed. For additional details, see Appendix B.

3.2 Distillation Datasets

To perform knowledge distillation, we utilize the Open Super-large Crawled Aggregated coRpus project (OSCAR) (Ortiz Su'arez et al., 2019), a large open-source corpus of raw unannotated web text. MLM pretraining, and by consequence Knowledge Distillation, requires a large amount of text data (Qiu et al., 2020) and OSCAR allows for the selection of a large amount of high-quality long-context text samples. This dataset is used during distillation alongside the commonly used training dataset, BookCorpus (Zhu et al., 2015). The selection of these two distillation datasets was determined through an experiment investigating the effect of different distillation datasets on downstream performance, as seen in Section 4.5.

When constructing our data to be used for knowledge distillation, we first filtered out all data from the OSCAR23.01 corpus which was not classified as having an eighty percent or higher chance of being English text to align with downstream tasks. To seek out only high-quality data, we also remove samples with quality annotations indicating tiny, short, or noisy sequences. We remove any data with a harmful perplexity score of 13.51 or less (Jansen et al., 2022) using perplexity scores provided by the OSCAR corpus (Ortiz Su'arez et al., 2019), and additionally remove any harmful categories. Finally, we select a sample from our filtered dataset to be used during distillation consisting of nearly three million sequences, then distil using this OSCAR subset alongside BookCorpus (Zhu et al., 2015), totaling 19 GB of uncompressed text.

3.3 GONERD

Data for GONERD (**G**iant **O**ak **NER D**ataset) was obtained by web scraping articles from publicly available sources such as online news and press release websites prior to being sampled and hand-labeled. A combination of automatic and manual filtering was then applied to remove text containing code and other unwanted data such as sequences deemed short, noisy, or duplicates.

As the explicit goal of GONERD is to gauge the performance of long-context NER models, we briefly quantify what sequence lengths are present within the dataset. We compare against CoNLL-

Length	CoNLL-2003 (512)	GONERD (4096)
mean	14.5	507.6
std. dev.	11.8	556.5
min	1.0	1.0
25%	6.0	170.0
50%	10.0	330.0
75%	22.0	658.0
max	124.0	6768.0

Table 1: Summary statistics on sequence length of CoNLL-2003 and GONERD. All statistics are computed over the whole dataset. "mean" and "std. dev." follow their usual definitions, "min" and "max" are the lengths of the shortest and longest datasets. "25%", "50%", "75%" are the 25th, 50th, and 75th percentiles, respectively.

2003, shown in Table 1, as it is widely used throughout NER literature. We find, on average, GONERD has much longer sequences than CoNLL-2003 (**507.6** vs. **14.5**), with a right skew as seen by the 50% percentile (**330**) being lower than the mean. We show this skew in Appendix A.2.

Furthermore, we find that approximately **35%** of GONERD sequences are above the 512 token threshold, whereas none of the CoNLL-2003 sequences occur in this range. Finally, we find that **0.2%** of sequences are longer than 4096, which are truncated at training and inference time. For more information on GONERD, including exploratory data analysis and additional comparisons with CoNLL-2003, see Appendix A.1 and A.2.

3.4 LSG RoBERTa Pretraining

Although the implementation of LSG RoBERTa (Condevaux and Harispe, 2023) is publicly available, there are currently no publicly accessible weights, neither compressed nor uncompressed, that have been pretrained on long-context sequences. While analysis on inference and memory utilization can be performed without these weights, undergoing a comprehensive performance analysis of LSG RoBERTa or utilizing this model in research or production requires further pretraining.

To address this issue, and to yield a pretrained teacher model as the first step towards developing a distilled student model, we pretrain a randomly initialized LSG RoBERTa model using the same dataset presented in LSG's inception (Condevaux and Harispe, 2023). This consists of English Wikipedia, BookCorpus, and CC_News.

4 Experiments

4.1 Inference Speed and Memory Usage

We calculate the average inference time and maximum GPU memory utilization for a variety of short-context and long-context transformer models as a proxy for predicting costs for hosting each model type in production, as displayed in Table 2. Moreover, we compare the potential cost of deploying efficient attention models versus their distilled equivalents. All models were tested in a uniform environment utilizing a single 80GB A100 GPU with a sequence length of either 512 or 4096 tokens and a batch size of 16.

	Model	Params (mil.)	Time (sec) 512	Time (sec) 4096	Mem. (MB) 512	Mem. (MB) 4096
Baseline	BERT$_{BASE}$	**109.5**	**0.135**	-	**4167**	-
	BERT$_{LARGE}$	335.1	0.379	-	5171	-
	RoBERTa	124.6	0.148	-	4843	-
	LegalBERT$_{BASE}$	**109.5**	**0.135**	-	**4167**	-
	XLM-R	278.0	0.237	-	11673	-
Compressed	DistilRoBERTa	82.1	0.089	-	4663	-
	DistilBERT	66.4	0.078	-	3987	-
	TinyBERT	**4.4**	**0.057**	-	**3033**	-
	MobileBERT	24.6	0.072	-	3639	-
	ALBERT	11.7	0.128	-	3783	-
Efficient	LSG	127.8	0.170	1.157	5472	23482
	➡	85.3	0.103	0.641	5292	23302
	Nyströmformer	111.2	0.159	1.866	4291	29059
	➡	**68.7**	0.090	0.787	4111	28879
	Longformer	148.7	0.149	1.110	4077	11881
	➡	95.5	**0.075**	**0.588**	**3857**	**11661**
	Big Bird	127.5	0.158	1.542	4938	26854
	➡	84.9	0.097	0.913	4757	26673

Table 2: Average Inference Speed and Peak GPU Memory Usage for sequence lengths of 512 and 4096. "➡" indicates distillation.

We find an average **45.2%** decrease in inference times for long-context efficient attention models and an average **2.6%** percent decrease in GPU memory utilization across all distilled efficient models. Among the distilled efficient students, Longformer produces the fastest inference speed and least peak GPU memory usage in both 512 and 4096 settings, despite having the most parameters.

We find that KD as discussed in Section 3.1 does not significantly impact peak GPU memory usage during inference across both efficient (LSG, Nyströmformer, Longformer, Big Bird) and non-efficient (DistilBERT, DistilRoBERTa) architectures. Larger modifications to the student architecture (TinyBERT, MobileBERT, ALBERT), produce varying speeds and levels of GPU memory usage.

Model	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Total
Metric	MCC	M/MM Acc.	Acc./F1	Acc.	Acc./F1	Acc.	Acc.	PCC/SRCC	Avg.
BERT[1]$_{BASE}$	52.1	84.6 / 83.4	88.9	90.5	71.2	66.4	93.5	85.8	79.6
BERT[1]$_{LARGE}$	60.5	86.7 / 85.9	89.3	92.7	72.1	70.1	**94.9**	86.5	82.1
RoBERTa[2]	**63.6**	**87.6**	**90.2**	**92.8**	**91.9**	**78.7**	94.8	**91.2**	**86.4**
LegalBERT	38.6	82.2 / 82.9	88.2	89.9	89.7	65.3	91.5	87.0 / 86.6	80.2
XLM-R	59.8	85.3 / 85.7	88.2 / **91.6**	92.3	90.7 / 87.6	77.3	93.3	90.9 / 90.6	86.1
DistilRoBERTa[2]	59.3	84.0	86.6	90.8	89.4	67.9	92.5	88.3	82.4
DistilBERT[1]	52.4	82.6	86.5	89.5	**88.6**	60.3	91.3	86.8	79.8
TinyBERT[1]	43.3	82.5 / 81.8	86.4	87.7	71.3	62.9	92.6	79.9	76.5
MobileBERT[1]$_{BASE}$	50.5	83.3 / 82.6	88.8	90.6	70.2	66.2	92.8	84.4	78.8
ALBERT	**59.8**	85.3 / 85.7	88.2 / **91.6**	92.3	90.6 / 87.7	**77.3**	92.9	**90.9 / 90.6**	86.1
LSG	59.8	86.7 / 86.1	89.7 / 92.5	**93.4**	89.8 / 86.3	70.0	94.8	90.2 / 90.0	84.2
➡	29.4	71.8 / 72.6	77.5 / 85.0	84.1	86.1 / 81.9	58.9	89.7	80.9 / 80.8	74.9
Nyströmformer	33.6	77.9 / 79.1	77.7 / 84.7	86.3	88.8 / 85.0	56.7	90.8	86.1 / 86.0	77.7
➡	43.4	78.6 / 78.6	75.2 / 83.8	85.8	89.3 / 85.9	58.5	90.8	86.2 / 85.8	78.5
Longformer	**61.3**	86.3 / 86.4	**91.9 / 94.2**	92.9	89.6 / 86.0	**77.6**	93.9	**90.8 / 90.5**	**86.8**
➡	55.5	82.0 / 81.8	82.1 / 86.9	87.7	**90.3 / 86.8**	54.2	91.7	86.2 / 86.0	80.9
Big Bird	51.6	**87.1 / 87.3**	87.8 / 91.3	91.0	**90.3 / 86.9**	68.2	**95.0**	86.5 / 86.5	84.1
➡	53.9	81.6 / 81.9	82.4 / 87.3	86.8	90.2 / 86.5	59.6	92.4	85.2 / 84.8	81.0

Table 3: Results on the validation set of the GLUE benchmark. "➡" indicates distillation performance. Results for [1,2] are pulled from MobileBERT (Sun et al., 2020) and DistilBERT (Sanh et al., 2019) papers, respectively; all other models are computed to completion. WNLI is not reported due to its problematic nature (Devlin et al., 2018).

4.2 Extending Input Sequence Length

To demonstrate the necessity of efficient attention architectures, we investigate the feasibility of using existing models on long-context tasks by allowing inefficient attention models to process longer sequence lengths. To explore this, as presented in Table 4, we examine the inference speed and peak GPU memory consumption during inference on full attention BERT-based models after being adjusted to compute sequence lengths of up to 4096 tokens, employing the same benchmarking methodology as seen in Section 4.1.

Model$^{\uparrow 4096}$	Time (sec)	GPU Mem (MB)
BERT$_{LARGE}$	5.806 (+423%)	39344 (+221%)
BERT$_{BASE}$	1.833 (+65%)	29886 (+152%)
RoBERTa	1.886 (+70%)	42506 (+258%)
DistilBERT	0.636 (+8%)	29706 (+155%)
DistilRoBERTa	0.798 (+36%)	42326 (+163%)
MobileBERT	1.274 (+117%)	**24406 (+109%)**
TinyBERT	**0.631 (+7%)**	29706 (+155%)
Longformer	1.110	11881
➡	**0.588**	**11661**

Table 4: Inference speed and GPU memory consumption when extending the maximum input sequence length from 512 to 4096 for various models. Percentages for non-compressed models are calculated against Longformer, while percentages for compressed models are calculated against distilled Longformer.

Our findings illustrate a noticeable trend: although it is possible to allow inefficient models to accept input sequences of up to 4096 tokens, there are significant speed and memory costs associated with doing so. Moreover, the newly initialized position embeddings would require anyone using these extended models to perform additional pretraining to yield acceptable long-context performance - a process that would be slower and more expensive than training an efficient attention model. This difficulty training would also inherently transfer to the process of fine-tuning these models on downstream tasks.

This evidence suggests that, although it is possible for full attention models to operate in long-context scenarios, it is often associated with increased inference and training costs when compared to non-distilled and distilled efficient attention models. As such, efficient attention models are an important step toward reducing the operational costs of long-context models, and distillation after conversion can be a useful methodology to further reduce costs and improve model accessibility.

4.3 Performance Benchmarks

GLUE We perform hyperparameter optimization using Population-Based Training (Jaderberg et al., 2017) on several baselines, augmented, efficient attention, and distilled efficient attention models on the GLUE benchmark (Wang et al., 2018). As seen in Table 3, we find that distilling efficient attention models yields compressed models capable of retaining, on average, **94.6%** of teacher model performance across all GLUE tasks and metrics. Distilled Big Bird produces the highest GLUE scores on average when compared to our distilled efficient attention models. Distilled Nystromformer sees a

slight increase in performance when compared to its teacher. Distilled LSG retains only **87.3%** of teacher performance.

Model	SQuAD1.1		HotpotQA		TriviaQA	
	EM	F1	EM	F1	EM	F1
BERT$_{BASE}$	80.97	88.21	-	-	-	-
BERT$_{LARGE}$	83.91	90.73	-	-	-	-
RoBERTa	**86.08**	**92.47**	-	-	-	-
LegalBERT	79.89	87.66	-	-	-	-
XLM-R	82.38	89.16	-	-	-	-
DistilRoBERTa	80.43	87.87	-	-	-	-
DistilBERT	77.01	85.21	-	-	-	-
TinyBERT	69.77	78.89	-	-	-	-
MobileBERT	80.83	88.56	-	-	-	-
ALBERT	**83.58**	**90.64**	-	-	-	-
LSG	80.61	87.89	56.96	72.11	47.34	51.82
➡	64.20	74.13	41.77	54.58	26.67	30.00
Nyströmformer	76.59	84.89	52.57	67.86	47.30	52.29
➡	70.87	80.51	48.54	63.87	44.55	49.68
Longformer	**85.92**	**92.24**	58.52	73.48	**55.29**	**60.52**
➡	77.93	85.81	49.86	64.96	46.75	51.42
Big Bird	84.94	91.44	**59.77**	**75.26**	54.29	59.33
➡	74.53	82.67	49.40	64.21	44.61	49.96

Table 5: SQuAD, HotpotQA, and TriviaQA Results.

Question Answering We train and evaluate all short-context and long-context transformer models on SQuAD1.1 (Rajpurkar et al., 2016). Moreover, we train and evaluate all long-context transformer models using a maximum sequence length of 4096 tokens on TriviaQA (Joshi et al., 2017) and Hot-potQA (Yang et al., 2018) for up to 5 and 10 epochs, respectively. Results are reported in Table 5.

We find that on SQuAD, HotpotQA, and TriviaQA, efficient attention students retained up to **94.8%**, **94.1%**, and **95.0%** of original model F1 performance, respectively. LSG RoBERTa was particularly strongly affected by the distillation process on long-context Question and Answering tasks, preserving **75.7%** of teacher performance on HotpotQA and **57.9%** on TriviaQA. Distilled Nyströmformer retains the most performance from its teacher with an average of **94.7%** across all QA benchmarks, but it is still outperformed by distilled Longformer by **2.3%**.

Named Entity Recognition We explore the impact of separately fine-tuning and evaluating both distilled and non-distilled efficient attention transformer models on CoNLL-2003 and GONERD in Table 6. We report each model's F1 performance on predicting Person (PER), Organization (ORG), Location (LOC), and Miscellaneous (MISC) tags. We find that performing knowledge distillation prior to fine-tuning on NER preserved **97.4%** of CoNLL-2003, while boosting GONERD F1 performance by **0.2%**.

Model	CoNLL-2003 (512)					GONERD (4096)				
	PER	ORG	LOC	MISC	ALL	PER	ORG	LOC	MISC	ALL
BERT$_{BASE}$	97.1	89.8	95.4	87.9	92.6	-	-	-	-	-
BERT$_{LARGE}$	**98.6**	**92.6**	**96.5**	**88.8**	**94.1**	-	-	-	-	-
RoBERTa	96.2	91.3	96.4	89.8	93.4	-	-	-	-	-
LegalBERT	95.3	87.2	94.8	86.0	90.8	-	-	-	-	-
XLM-R	95.6	90.2	95.8	89.8	92.9	-	-	-	-	-
DistilRoBERTA	**96.7**	**92.1**	**96.7**	**90.1**	**93.9**	-	-	-	-	-
DistilBERT	96.7	89.2	95.4	88.3	92.4	-	-	-	-	-
TinyBERT	95.6	87.8	95.3	86.8	91.4	-	-	-	-	-
MobileBERT	97.8	90.2	96.4	87.9	93.1	-	-	-	-	-
ALBERT	93.8	85.6	94.5	86.3	90.1	-	-	-	-	-
LSG	96.6	90.0	95.2	88.1	92.5	**76.5**	66.7	64.0	**78.7**	70.2
➡	89.8	80.0	92.2	81.3	85.8	69.8	59.0	60.7	72.6	64.1
Nyströmformer	94.8	85.1	93.3	86.4	89.9	72.4	59.5	59.6	75.1	65.0
➡	95.3	85.1	94.2	85.6	90.1	70.6	56.4	60.2	70.5	63.3
Longformer	96.2	**91.5**	**96.8**	90.5	**93.8**	75.9	**68.0**	65.1	77.3	**70.6**
➡	**96.7**	91.2	96.7	89.8	93.6	71.8	65.1	63.3	76.3	67.7
Big Bird	96.4	91.8	96.4	89.8	93.6	75.9	65.4	**66.3**	73.1	69.8
➡	96.2	90.4	96.2	89.7	93.1	71.6	63.2	61.7	73.2	66.4

Table 6: **N**amed **E**ntity **R**ecognition (**NER**) F1 Performance on CoNLL-2003 and GONERD.

4.4 Evaluating the Effect of Convert and Distill on Downstream Performance

Model	Inf. Time (sec)	GPU Mem.	GLUE	SQuAD1.1	CoNLL-2003	HotpotQA	TriviaQA	GONERD
RoBERTa	.148	4843	86.35	92.47	93.43	-	-	-
Δ KD	-39.9%	-3.7%	-4.6%	-5.0%	+0.5%	-	-	-
Δ *Convert*	+0.7%	-15.8%	-0.5%	-2.5%	+0.3%	73.48	60.52	70.6
Δ *Convert*+KD	**-49.3%**	**-20.4%**	**-6.3%**	**-7.0%**	**+0.2%**	**-11.6%**	**-15.0%**	**-4.1%**

Table 7: Effects of introducing Knowledge Distillation and Longformer attention into RoBERTa on various tasks. We report average score for GLUE and overall F1 for QA and NER. "Δ" indicates a change from the base model. Results are compared against RoBERTa on short-context tasks and against Longformer (Δ *Convert*) on long-context tasks. Sequence lengths of 512 are used for inference time and memory usage.

To gauge the contribution of each component of the **Convert-Then-Distill** pipeline, we provide the computational cost and performance with respect to RoBERTa after undergoing conversion and distillation in Table 7. In contrast to Table 4, the inference speeds and max GPU memory usages are calculated on sequences of up to 512 tokens. Within this range, we see that KD greatly improves inference speed while resulting in a minor decrease in maximum GPU memory utilization. Conversely, we see conversion to an efficient attention mechanism (in this case, Longformer) yields significant decreases in maximum GPU memory utilization and minor improvements in inference speed. Together, we find that *Convert*+KD is additive in its effects: per-

forming Conversion and KD yields models with both improved inference speeds and reduced GPU memory requirements,

We find long-context QA performance is heavily degraded by introducing *Convert*+KD into training in comparison to other tasks, whereas conversion does not significantly impact performance. However, long-context NER appears to be an exception, as introducing *Convert*+KD into GONERD has a significantly lower impact on performance. Finally, we note that the distillation process as used in DistilBERT (Sanh et al., 2019) and detailed in Appendix B leaves room for further improvement: developing distillation methods tailored for individual efficient attention mechanisms, tasks, and architectures may yield improved performance.

4.5 Evaluating the Effect of Distillation Data on Downstream Performance

Distillation Data	GLUE	SQuAD1.1	HotpotQA	CoNLL-2003	GONERD
BC + WIKI	75.6	77.9	57.7	92.2	65.5
OSCAR + BC	**78.9**	**85.8**	**65.0**	**93.6**	**67.7**
OSCAR + WIKI	77.1	78.8	60.6	92.8	66.3
WIKI	68.7	76.5	57.9	91.9	63.8
OSCAR	60.5	25.4	39.6	73.5	38.3
BC	72.9	56.4	40.7	90.4	40.0

Table 8: Effects of choice of data on KD performance using Longformer RoBERTa with a train batch size of 4. Average score, not including WNLI, is reported for GLUE and overall F1 is reported for QA and NER. A full expansion of GLUE results is given in Table 11.

For a more comprehensive evaluation of our knowledge distillation process, we report the performance of Longformer-RoBERTa after distillation on various permutations of the OSCAR, BookCorpus, and English Wikipedia datasets, as seen in Table 8.

We find that, although OSCAR+BookCorpus yields the best performance on both short-context and long-context tasks, the performance gap between OSCAR+BookCorpus, Wikipedia+BookCorpus, and OSCAR+Wikipedia is very modest. However, as OSCAR+BookCorpus proved to be the most performant, we utilize this dataset when distilling efficient attention models.

5 Conclusion

In this work, we performed an investigation into the **Convert-Then-Distill** paradigm, the process of (1) *converting* a teacher model to utilize an efficient attention mechanism, (2) pretraining the converted teacher model, (3) distilling into a smaller student model, then (4) fine-tuning the student on a downstream task. We saw an average decrease in inference times of up to **58%**. The efficient attention students preserved up to **98.6%** of performance across short-context (GLUE, SQuAD, CoNLL-2003) tasks, **94.1%** of HotpotQA performance, **95.0%** of TriviaQA performance, and **97.4%** percent of GONERD performance when compared to their teacher models. We saw distilled Nyström-former retained the most performance when compared to its teacher, while distilled Longformer had the best base performance across most tasks. We introduced GONERD, a long-context NER dataset consisting of large amounts of hand-labeled web text data. Finally, we release all models on the Hugging Face Hub for general use. Our research demonstrates that, for most models on most tasks, employing knowledge distillation on efficient attention architectures can be a highly effective approach. This technique yields models with a high level of performance on both short and long-context tasks at a fraction of the cost.

Limitations

As seen in Section 4.3, we find that, in both short and long sequences, **Convert-Then-Distill** degrades performance to a greater extent than either Convert or KD separately. This performance degradation warrants further investigation into generalization capabilities of efficient students.

Following this, many distillation procedures have been proposed since the original technique of DistilBERT (Sanh et al., 2019). Using more recent distillation methods, or developing distillation methods tailored toward an individual efficient attention architecture, may decrease the student-teacher performance gap and increase generalizability.

Our work is constrained to the **Convert-Then-Distill** paradigm which, although intuitive, is not obviously better than **Distill-Then-Convert** or other alternatives. For instance, it may be possible that non-efficient teachers produce better students which can then be extended to the 4096 or greater token range. Further investigation into the optimal method for developing distilled efficient attention models may be necessary to further close the aforementioned performance gap.

Finally, GONERD suffers from a domain bias as it is composed entirely of news-like webtext data and commonly littered with legal jargon. We attempt to control for this bias by comparing with LegalBERT and ablating on choices of pretraining data, but we note this bias for any potential users of GONERD. For general long-context NER use, additional pretraining may be required.

References

Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. 2015. Practical and optimal lsh for angular distance. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Charles Condevaux and Sébastien Harispe. 2023. Lsg attention: Extrapolation of pretrained transformers to long sequences. In *Advances in Knowledge Discovery and Data Mining*, pages 443–454, Cham. Springer Nature Switzerland.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. 2017. Population based training of neural networks.

Tim Jansen, Yangling Tong, Victoria Zevallos, and Pedro Ortiz Suarez. 2022. Perplexed by quality: A perplexity-based method for adult and harmful content detection in multilingual heterogeneous web data.

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, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

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*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Pedro Javier Ortiz Su'arez, Benoit Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Chal-

lenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut f"ur Deutsche Sprache.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

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*.

Or Sharir, Barak Peleg, and Yoav Shoham. 2020. The cost of training nlp models: A concise overview.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. 2021. Nyströmformer: A nyström-based algorithm for approximating self-attention.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, Hao Peng, Jianxin Li, Jia Wu, Ziwei Liu, Pengtao Xie, Caiming Xiong, Jian Pei, Philip S. Yu, and Lichao Sun. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

A GONERD

A.1 Data Collection

Data for GONERD was obtained through Giant Oak's GONER software, which scraped web articles from public facing online news sources as well as the U.S. Department of Justice's *justice.gov* domain. This webtext data was randomly sampled with an upweighted probability toward documents from *justice.gov* so that *justice.gov* consisted of roughly 25% of the total GONERD dataset. A combination of automatic and manual filtering was then applied to remove text containing code and other unwanted data, such as sequences deemed to be short, noisy, or near-duplicates.

A.2 Exploratory Data Analysis

Type	CoNLL-2003 (512)			GONERD (4096)		
	#	p	p^1	#	p	p^1
O	251k	.832	-	1013k	.896	-
B-PER	10k	.033	.198	24.4k	.022	.200
I-PER	6.9k	.023	.138	22.9k	.020	.188
B-ORG	9.3k	.031	.184	21.5k	.019	.177
I-ORG	5.3k	.018	.104	23.0k	.020	.188
B-LOC	10.6k	.035	.210	16.4k	.014	.134
I-LOC	1.7k	.006	.033	9.7k	.009	.079
B-MISC	5.1k	.017	.100	2.4k	.002	.020
I-MISC	1.7k	.006	.034	1.7k	.002	.014
Total	301k	1.0	1.0	1131k	1.0	1.0

Table 9: Occurrence of PER/ORG/LOC/MISC/O tags in CoNLL-2003 and GONERD. p represents the proportion of a tag over the total amount of labeled tokens and p^1 represents the proportion over non-O tokens.

Sequence Length Distribution In Figure 1, we display the distribution of CoNLL-2003 sequences in orange and GONERD sequences in blue. To produce the figure, we used standard Kernel Density Estimation (KDE) through the *kdeplot* function of the Python seaborn library. For the GONERD distribution, we used the default parameters of the *kdeplot* function, but for CoNLL-2003, we used a higher KDE bandwidth and upsampled the distribution in the 256 range, thereby giving CoNLL-2003 a slightly synthetically higher distribution, resulting in CoNLL-2003 sequences appearing to be longer than they actually are. We perform this to account for to the extreme gap in average sequence length between CoNLL-2003 and GONERD. CoNLL-2003 has a large number of short sequences which make the table significantly taller, making visually comparing their distributions unintelligible. We briefly provide summary statistics in Table 1 to evidence this gap.

Entity Makeup For our NER task, we evaluated the distribution of tags to gain a deeper understanding of our evaluation results. As seen in Tables 9 and 10, although ConLL-2003 consists of more sequences, GONERD has $3.5\times$ as many labeled tokens. Additionally, we find that names in GONERD tend to be longer than CoNLL-2003, as evidenced by the proportion of I to B tokens across all NER tags. For GONERD, we find this proportion to be 57.3/64.7 in comparison to 15.6/35 for CoNLL-2003.

As seen in Table 6, LOC and ORG are the most difficult for both teacher and student teachers

Website	#	pdf	cdf
justice.gov	542	.242	.242
ctvnews.ca	190	.085	.327
msn.com	146	.065	.392
southcarolinapublicradio.org	54	.024	.416
express.co.uk	41	.018	.434
dailyrecord.com	33	.015	.449
dailyvoice.com	28	.013	.462
nbcnews.com	23	.010	.472
newsbreak.com	21	.009	.481
chicagotribune.com	19	.008	.489
...	...	...	...
Total	2237	1.0	1.0

Table 10: Occurrence of domains in GONERD. "#" is the raw amount of samples occuring under a domain, "pdf" is the proportion of samples in the whole dataset for that domain, and "cdf" is the cumulative proportion of samples sorted by frequency. Results are sorted by descending "pdf." Asteriscs indicate data not shown.

to learn in GONERD. This may come as a surprise when considering the MISC tag, in which all efficient attention models obtained better performance despite MISC's fewer samples. One possible explanation for the discrepancy in MISC performance is in how GONERD handles MISC labeling. GONERD has a fixed schema for MISC: ages, addresses, and phone numbers, while everything else is not marked as a valid entity. As this reduces the diversity of this category, this could make the MISC tag easier for models to learn to detect. This is in stark comparison to CoNLL-2003, in which MISC consists of adjectives and events, making it a very diverse category. This can be evidenced by the performance difference for efficient attention models on CoNLL-2003, where MISC was the most difficult tag for models to learn. Outside of the discrepancy on MISC, GONERD's makeup closely resembles ConLL-2003 regarding the distribution of non-O tags but leverages long-context, making it a valuable asset to long-context NER models.

Domains In Table 10, we show the frequencies of the top ten domains occurring in GONERD, ranked by relative occurrence. The raw number of samples under a domain is denoted by "#", the relative proportion by "pdf," and the cumulative by "cdf."

Aligning with expectations, we see that *justice.gov* appears in **24.2%** samples, a website full of news and legal language, primarily in the form of criminal charges and sentencing. However, as
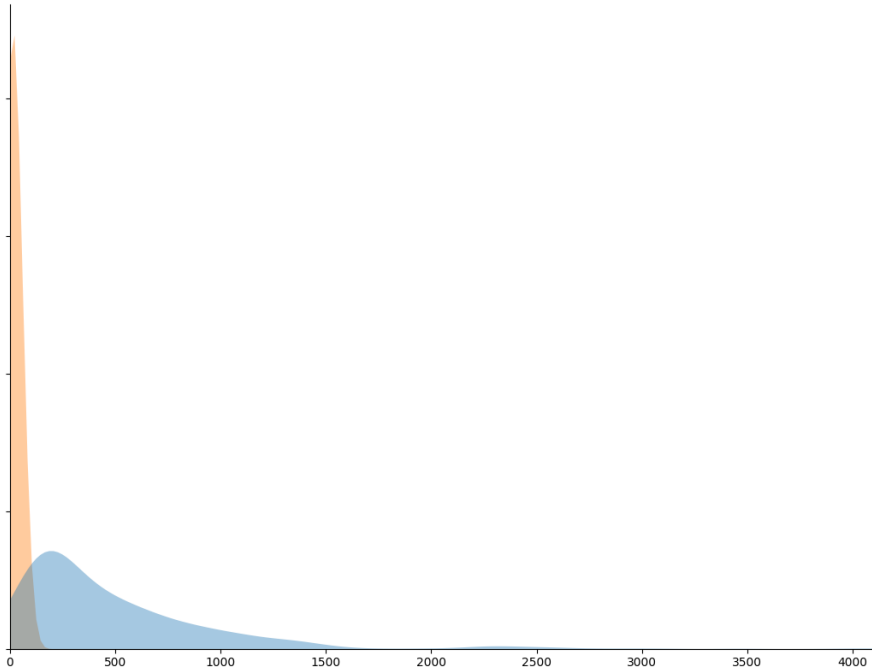
Figure 1: Sequence length distribution using Kernel Density Estimation (KDE) of CoNLL-2003 (orange) and GONERD (blue). Smoothing was performed with Gaussian KDE using the seaborn kdeplot function. x-axis is number of tokens and y-axis is probability density.

domains progress, the relative contribution drops off exponentially, with the top ten domains only making up **48.9%** of GONERD whereas there are 369 domains present within the dataset.

B   Knowledge Distillation Details

Briefly, we give an overview of the student objective used in our distillation experiments, which we frame as the linear combination of *supervised training* loss, *distillation* loss, and *hidden state* loss. Our supervised training loss is the standard masked language modeling loss (Devlin et al., 2018). Our distillation loss is a cross entropy over soft targets (Hinton et al., 2015; Sanh et al., 2019), which are calculated by applying a softmax with temperature to the output logits:

$$p_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

where p_i is the probability of logit z_i and T is temperature, which controls the smoothness of the distribution. Following the methods outlined in DistilBERT (Sanh et al., 2019), we use a cosine embedding loss between the hidden states vectors of the teacher and student as a hidden state loss. Our overall training objective can thus be written

as

$$\mathcal{L}_{student} = \alpha\mathcal{L}_{mlm} + \beta\mathcal{L}_{ce} + \gamma\mathcal{L}_{cse}$$

We take $\alpha = 2.0$, $\beta = 5.0$, $\gamma = 1.0$, and $T = 2.0$. Finally, we train the student by minimizing the associated empirical risk with the AdamW optimizer.

C   Data Ablation Results

Finally, we expand upon the GLUE performance given in Table 8, distilling Longformer RoBERTa on various permutations of the BookCorpus (BC), English Wikipedia (ENW), and OSCAR datasets and evaluating on all GLUE tasks. All models are trained identically as given in Section 4.5.

We find that distilling Longformer on OSCAR and BookCorpus yields the highest GLUE scores, with an average of **78.9** across all tasks and metrics. However, both BookCorpus and English Wikipedia as well as OSCAR and English Wikipedia still yield very similar results, with the most notable differences being in the CoLA and MNLI tasks. We see significantly lower scores, particularly on CoLA, when Longformer is distilled using only short or long sequences. This indicates that it may be necessary for efficient attention models to be distilled using a mixture of both short and long-context data to ensure maximum student performance.

Model	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Total
Metric	MCC	M/MM Acc.	Acc./F1	Acc.	Acc./F1	Acc.	Acc.	PCC/SRCC	Avg.
BC + ENW	41.7	77.3 / 77.6	82.6 / 87.6	83.9	88.0 / 84.7	56.7	89.7	84.4 / 84.1	75.6
OSCAR + BC	**52.1**	**81.8 / 82.3**	**84.8 / 88.9**	**87.3**	**89.9 / 86.6**	57.0	**91.7**	**86.3 / 86.1**	**78.9**
OSCAR + ENW	46.1	76.8 / 78.9	83.8 / **88.9**	86.2	87.9 / 83.1	**58.5**	91.3	85.2 / 84.9	77.1
ENW	7.1	73.8 / 74.3	79.4 / 86.0	81.6	86.0 / 80.9	54.2	85.1	81.6 / 81.3	68.7
OSCAR	10.6	67.7 / 44.0	72.3 / 82.0	75.3	82.6 / 77.5	47.3	81.5	56.0 / 56.8	60.5
BC	38.7	74.2 / 75.6	75.7 / 83.6	83.1	87.1 / 82.5	55.2	88.3	78.6 / 78.3	72.9

Table 11: Full validation results for GLUE on the students in the distillation ablative experiment in Section 4.5.