Compression Methods for Transformers in Multidomain Sentiment Analysis

1st Wojciech Korczyński
Department of Artificial Intelligence
Wrocław University of Science and Technology
Wrocław, Poland
wojciech.korczynski@pwr.edu.pl

2nd Jan Kocoń
Department of Artificial Intelligence
Wrocław University of Science and Technology
Wrocław, Poland
jan.kocon@pwr.edu.pl

***Abstract*—Transformer models like BERT have significantly improved performance on many NLP tasks, e.g., sentiment analysis. However, their large number of parameters makes real-world applications difficult because of computational costs and latency. Many compression methods have been proposed to solve this problem using quantization, weight pruning, and knowledge distillation. In this work, we explore some of these task-specific and task-agnostic methods by comparing their effectiveness and quality on the MultiEmo sentiment analysis dataset. Additionally, we analyze their ability to generalize and capture sentiment features by conducting domain-sentiment experiments. The results show that the compression methods reduce the model size by 8.6 times and the inference time by 6.9 times compared to the original model while maintaining unimpaired quality. Smaller models perform better on tasks with fewer data and retain more remarkable generalization ability after fine-tuning because they are less prone to overfitting. The best trade-off is obtained using the task-agnostic XtremeDistil model.**

***Index Terms*—knowledge distillation, transformers, sentiment analysis, MultiEmo**

I. Introduction

Large pre-trained language models (PLMs) have become the most popular research focus in the field of Natural Language Processing (NLP) [1]–[4]. Since the creation of the transformer architecture [5], most of the best models have been based on it. They are pre-trained on a large unsupervised text corpus and then fine-tuned on a downstream task. These models have achieved outstanding results in many Natural Language Understanding (NLU) tasks (e.g., GLUE [6]).

One of the NLP tasks is sentiment analysis, which has become very popular recently [7], [8]. Emotion and sentiment perception of texts can be used for many purposes, e.g., prediction of election results [9], and detection of threats in future events [10]. The analysis of customer reviews and opinions is also of great interest [11], [12] because it can help manufacturers create better products. Such applications show a need for effective and efficient methods to perform sentiment analysis of the text. Additionally, these methods should be able to work in a multidomain manner at the level of single sentences and full documents.

The transformer-based PLMs can obtain great results in sentiment analysis [13]. Nevertheless, their efficiency is very low due to a large number of parameters. Long inference time and big size cause difficulties in deploying them in real-life environments. There are a few techniques that aim to compress large-scale models, including quantization [14], weight pruning [15], and knowledge distillation, KD [16]. In the context of transformer models, apart from some works concerning quantization [17]–[19] and weight pruning [20]–[22] the most attention was put on KD [23]–[27]. Recent research focuses on hybrid methods that integrate quantization, pruning, and KD [28]–[31].

Due to abundantly available methods, it can be hard to choose the most appropriate one concerning a given application, e.g., sentiment analysis. The best approach does not need to be the one with the highest quality measures. In the context of compression methods, evaluation, speedup, and size reduction are equally important issues. Training time can also be a relevant factor.

This work presents a thorough analysis of compression methods for $\text{BERT}_{\text{BASE}}$. We test them on MultiEmo sentiment analysis task [32], considering classification quality and work efficiency. Our experiments focus on sentence and document level granularity. Domain issue is also considered. As a result, it turns out that the best trade-off between efficiency and quality is obtained for the XtremeDistil method, which is 6.4x faster, 8.6x smaller, and retains most of the original model's quality.

II. Related Work

A. Model Compression.

Compression of machine learning models has been studied for a long time. For example, quantization methods reduce the number of bits needed to represent parameters in a model [14]. Another approach is network pruning [15] which removes redundant parameters or components. It can be done in an unstructured manner where individual, less important weights are pruned or in a structured manner where entire blocks of weights are pruned at once.

This work was financed by (1) the National Science Centre, Poland, project no. 2019/33/B/HS2/02814 and 2021/41/B/ST6/04471; (2) the Polish Ministry of Education and Science, CLARIN-PL; (3) the European Regional Development Fund as a part of the 2014-2020 Smart Growth Operational Programme, CLARIN – Common Language Resources and Technology Infrastructure, project no. POIR.04.02.00-00C002/19; (4) the statutory funds of the Department of Artificial Intelligence, Wroclaw University of Science and Technology.

2375-9259/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDMW58026.2022.00062

KD is a procedure where knowledge of a large model (teacher) is transmitted into a compact model (student) so that the student mimics the teacher's behavior. Bucila et al. [33] compress a large and cumbersome ensemble model into a single neural network while preserving the performance by minimizing the mean squared error (MSE) between the outputs of both models. Hinton et al. [16] distill an ensemble of neural nets into a single neural net using so-called soft targets from the smoothed teacher outputs and the hard targets from the data. Romero et al. [34] shows that the performance of the student model can be further improved by using not only the teacher outputs but also its internal representation.

Since that time, many attempts have decreased the performance gap between the teacher and the student or compressed the model significantly. These attempts include weight quantization of the student [35], or a multistep KD with an assistance of an intermediary teacher (Teacher Assistant KD) [36]. It bridges the size gap between the teacher and the student model.

B. Language Models Pretraining.

Pretraining has been widely applied in NLP. The feature-based approach focuses on learning word representations (word embeddings). These representations can be context-independent as word2vec [37], GloVe [38], fastText [39] or contextualized as ELMo [1].

Since the emergence of transformers [5], a fine-tuning approach has been developed – a transformer language model is firstly pretrained on a large corpus in an unsupervised manner and afterward fine-tuned on some downstream tasks using labeled data. It has caused significant performance improvements in many natural language understanding (NLU) tasks. Many popular models are based on that framework, such as GPT [3], BERT [2], and ELECTRA [4].

C. Transformer Model Compression.

One of the limitations of transformer models like BERT is their large size and long time needed for training or inference. That is why compressing these models has become a very critical issue.

First quantization methods applied on transformer models reduce weights to 8 bits, e.g., Q8BERT [17]. The next attempts successfully quantize parameters to 2 bits (ternariztion in TernaryBERT [40] with 14.9x compression rate) or even 1 bit (binarization in BinaryBERT [19] with 24x compression rate).

Another approach for compressing transformer models is pruning. One example of unstructured pruning is magnitude weight pruning [20] which removes weights with values below a certain threshold. The experiments show that reducing 30-40% of weight does not affect performance significantly. Other works focus on structured pruning. Poor Man's BERT [21] simply drops the chosen transformer layers obtaining the best results by removing the top layers. BERT-of-Theseus [22] performs the training of the BERT model with random replacement of the modules with more compact ones.

One of the first attempts to compress the BERT model using KD was Patient KD (PKD) [23]. It is a task-specific method that transfers the knowledge from the predictive and intermediary layers. The performance of *student* models is better than that of the models with the same architecture but solely fine-tuned. FastBERT [41], and RomeBERT [42] are self-KD methods that ensure adaptive inference time by a dynamic mechanism of early exits from the shallower layers.

Task-agnostic KD methods are universal because they allow for fine-tuning on any NLU task. One of such methods is DistilBERT [24] where distillation is applied at the pretraining stage on a large corpus using soft targets and cosine embedding loss. DistilBERT requires that the student's hidden dimension be the same as the teacher's. It is not necessary for TinyBERT [25] and MiniLM [26]. TinyBERT is a two-staged method that allows obtaining a general task-agnostic or task-specific model. The knowledge is distilled from the predictive layer, hidden layers, self-attention matrices, and the embedding layers. MiniLM proposes to use only self-attention values of the last transformer layer during KD, showing that they are fundamental components in transformers. It also benefits from the Teacher Assistant KD approach. Another task-agnostic method is MobileBERT [27] which compresses the model width.

Many recent works focus on integrating various techniques to compress BERT models. DynaBERT [29] combines structured weight pruning and KD, allowing a task-specific model whose size can be dynamically adjusted. LadaBERT [28] and ROSITA [31] are task-specific methods which utilize weight pruning, matrix factorization and KD. XtremeDistilTransformers [30] is a task-agnostic framework which combines KD and matrix factorization. It is not trained on an unsupervised task but some source tasks. To improve the results, data augmentation and progressive learning are utilized.

Many compression methods for the BERT model are empirically compared by Ganesh et al. [43]. However, it is done using the results on GLUE and SQuAD [44] reported by the authors of the original papers. On the other hand, in our work, we compare the chosen compression methods by conducting experiments on MultiEmo, which is a completely different sentiment analysis task. Additionally, it allows to take into account text level and domain aspects for further analysis.

III. MULTIEMO DATASET

MultiEmo [32] is a benchmark dataset for the sentiment analysis task. This collection is based on the PolEmo 2.0 dataset [45], [46], consisting of more than 8,000 consumer reviews, containing more than 57,000 sentences. These opinions cover 4 different domains: medicine, hotels, school, and products. The dataset is provided with a priori split into training, development, and test sets in ratio 80%/10%/10%.

Within PolEmo 2.0, all opinions and all sentences forming them were annotated with the sentiment. 3 independent experts annotated each item. The sentiment classes used were: positive, negative, neutral, and ambivalent. Annotator agreement measured using Positive Specific Agreement for texts was

over 90%, while for sentences, it was over 87%. The original PolEmo dataset was created for Polish texts.

The MultiEmo dataset contains PolEmo opinions automatically translated into ten languages using DeepL[1]. Both PolEmo 2.0[2] and MultiEmo[3] are available under an open license. In this work, we use the English translations to compare compression methods.

IV. Compared Methods

In the work, we compare state-of-the-art methods compressing the $\text{BERT}_{\text{BASE}}$ model. They are either task-specific or task-agnostic. Most of them use knowledge distillation, but some benefit from quantization and pruning as well. Table I presents them. The following methods are considered:

- **DistilBERT** [24] is a task-agnostic method which distills knowledge at the pre-training stage on a large unsupervised corpus. As a loss function, it uses soft targets distillation loss and cosine embedding loss. The student is initialized from the layers of the teacher $\text{BERT}_{\text{BASE}}$ model. We use the available pretrained DistilBERT_6 which is fine-tuned for MultiEmo tasks.
- **XtremeDistilTransformers** [30] is a task agnostic method which distills knowledge on a source task. It uses matrix factorization and KD techniques.
  The embedding matrix is compressed using Singular Value Decomposition (SVD). A training objective in KD consists of MSE for hidden states, self-attention states, task-specific logits, and cross-entropy (CE) loss for ground truth data. A learnable linear transformation on the hidden student states is performed to match the teacher's hidden dimensions. For better performance, the source task data is augmented. We fine-tune the available XtremeDistil student model with $L = 6$ layers and hidden dimension $H = 256$. It was initialized with the MiniLM model and pretrained on the MNLI task with $\text{ELECTRA}_{\text{BASE}}$ [4] as a teacher (instead of $\text{BERT}_{\text{BASE}}$), which has the same architecture as $\text{BERT}_{\text{BASE}}$ but was pre-trained using different tasks, therefore features slightly better performance.
- **TinyBERT** [25] is a method that performs KD in both the pretraining and the task-specific fine-tuning stage. The loss consists of MSE between the hidden states, the embedding layer, the self-attention layers, and CE between the outputs. The dimensions are aligned by learnable linear transformations.
  A large text corpus is used in the first stage, called general distillation. It produces a general TinyBERT model, the student in the second stage where distillation on some downstream augmented data is performed.
  In our work, TinyBERT is treated either as a task-agnostic method (denoted as $\text{TinyBERT}_{k,\text{TA}}$) where a general model is directly fine-tuned or as a task-specific one (denoted as $\text{TinyBERT}_{k,\text{TS}}$) with a further distillation on a downstream MultiEmo task. That distillation is two-phased: (1) for the intermediary layers and (2) for the predictive layer. We use available general TinyBERT_4 ($H = 312$), TinyBERT_6 ($H = 768$) models which were pre-trained with $\text{BERT}_{\text{BASE}}$ as a teacher.
- **BERT-of-Theseus** [22] is a task-specific, structured pruning method. During training, the original modules of the large model (predecessor) are progressively replaced with smaller modules. The probability of replacement increases linearly as the training procedure continues. In the second stage, the successor model consisting only of the smaller modules is additionally fine-tuned. In the first stage, the weights of the predecessor are frozen; in both stages, CE loss on the ground truth data is used. In our work, the predecessor model is $\text{BERT}_{\text{BASE}}$ which is compressed to BERT_6 by replacing the subsequent two layers with one layer.
- **ROSITA** (**R**efined BERT c**O**mpre**S**sion with **I**n**T**egr**A**ted techniques) [31] is a task-specific method which uses weight pruning, matrix factorization, and KD. The embedding layer is reduced with the SVD method, and the last transformer layers, less important neurons of transformer layers, and attention heads are pruned. It is performed in a progressive way with interleaving KD steps. KD loss consists of CE between the outputs and MSE for the hidden states and the embedding layers.
  The process is divided into three stages: (1) distillation of the intermediary student model with the same architecture as the teacher; (2) iterative depth pruning with KD (as a result, a model with fewer layers is obtained); (3) iterative width pruning where attention heads are pruned, and dimensions of hidden states and embeddings are decreased, pruning interleaves KD. During the entire process, the augmented data for a task are used. In our experiments, $\text{BERT}_{\text{BASE}}$ is used as the teacher; the final model has 8 layers, 2 self-attention heads, and a hidden dimension of 128.

V. Experiments

In this section, we conduct various experiments which compare how the chosen compressing methods work for the MultiEmo sentiment analysis task in the context of effectiveness and quality. Additionally, the analysis focuses on a domain aspect of texts. We test the quality of the compressed models for both the single domain (SD) with MultiEmo (Hotels, Medicine, Products, School) as well as the leave-one-domain-out (DO) scenario. These scenarios are tested at the level of whole opinion texts (SDT, DOT) and the level of individual opinion sentences (SDS, DOS) in the English language.

We evaluated DistilBERT, XtremeDistilTransformers, TinyBERT, ROSITA, and BERT-of-Theseus. For reference, we also fine-tune $\text{BERT}_{\text{BASE}}$ for 4 epochs with learning rate 5e-5 and weight decay 0.01. We fine-tune task-agnostic compressed models with the same parameters as for $\text{BERT}_{\text{BASE}}$.

The best performing fine-tuned $\text{BERT}_{\text{BASE}}$ model serves as a teacher in the task-distillation TinyBERT and ROSITA and as

[1]https://www.deepl.com/
[2]https://clarin-pl.eu/dspace/handle/11321/710
[3]https://clarin-pl.eu/dspace/handle/11321/798

TABLE I
COMPARISON OF THE ANALYZED COMPRESSION METHODS.

Method	Task-agnostic	Embedding Layer	Hidden state	Attention state	The same hidden dimension	Data Augmentation	Weight Pruning
DistilBERT	✓	✓($\mathcal{L}_{\cos}$)			✓		
XtremeDistil	✓	✓(SVD)	✓	✓		✓	
TinyBERT	✓/ ✗	✓(MSE)	✓	✓		✓	
BERT-of-Theseus					✓		✓
ROSITA		✓(SVD)	✓			✓	✓

TABLE II
COMPARISON OF THE CONSIDERED COMPRESSION METHODS FOR ALL MULTIEMO DOMAINS AT THE SENTENCE LEVEL. THE RESULTS ARE AVERAGED ON FIVE REPETITIONS.

Method	Parameters	Memory [MB]	Training [min]	Eval [s]	Accuracy	Macro F1	Macro Recall	Macro Precision
$\text{BERT}_{\text{BASE}}$	109M	418	26.0	13.9	78.8	74.7	74.1	75.8
DistilBERT	67M	255 (1.6x)	14.1	7.0 (2.0x)	77.9	73.9	73.3	**74.8**
XtremeDistil	**13M**	**49 (8.6x)**	6.5	2.3 (6.2x)	76.7	72.4	71.7	73.9
$\text{TinyBERT}_{6,\text{TA}}$	67M	255 (1.6x)	14.1	7.2 (1.9x)	77.7	73.8	73.5	74.3
$\text{TinyBERT}_{6,\text{TS}}$	68M	258 (1.6x)	62.9	7.2 (1.9x)	**78.2**	**74.4**	**74.1**	74.7
$\text{TinyBERT}_{4,\text{TA}}$	14M	55 (7.6x)	**5.5**	2.1 (6.6x)	76.3	72.0	71.1	73.7
$\text{TinyBERT}_{4,\text{TS}}$	15M	56 (7.5x)	38.3	**2.0 (6.9x)**	76.3	72.2	71.5	73.4
BERT-of-Theseus	67M	255 (1.6x)	19.1	7.8 (1.8x)	76.6	72.5	72.1	73.2
ROSITA	14M	55 (7.6x)	121.5	3.4 (4.1x)	77.5	72.9	72.2	74.9

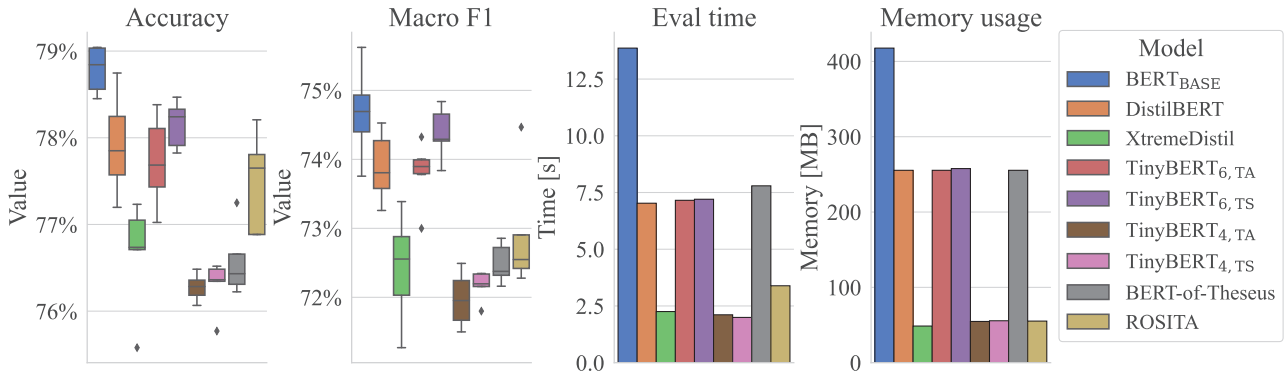

Fig. 1. Comparison between the considered compression methods applied for the sentence level MultiEmo data from all domains. The results are averaged on five repetitions.

a predecessor in BERT-of-Theseus. For TinyBERT, a student model is a general TinyBERT model. Both two distillation phases are performed for 4 epochs with a learning rate of 5e-5 and weight decay of 0.01. For training BERT-of-Theseus, these parameters are the same. The initial replacing rate is set to 0.3, and the coefficient k of the linear scheduler controlling the current value of replacing rate is 0.00014. For ROSITA, all three stages of the compression process are performed for four epochs with a weight decay of 0.01. The learning rate for the first and the second stage is 2e-5, and the third one is 5e-5. The proportion of pruning steps in the second stage is 0.2 and in the third stage is 0.1.

In each case, the model is trained by minimizing the loss related to a given model/method on the train data set using the Adam optimizer [47] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For testing, the model checkpoint is used for which the best results on the development set were obtained. The batch size is 16 for the sentence-level tasks and 8 for the document-level ones. The maximum sequence length is 128 and 256, respectively, for the sentence and the document level. All experiments are repeated 5 times. They are conducted on GeForce GTX 1080 Ti. Data augmentation, originally utilized for TinyBERT and ROSITA, was not performed during our experiments.

Results of the experiments performed at the sentence level are presented in Table II and Figure 1. These results demonstrate that: (1) The best results are obtained for $\text{TinyBERT}_{6,\text{TS}}$ with retained 99.2% of accuracy. (2) The most efficient models are XtremeDistill, $\text{TinyBERT}_{4,\text{TA}}$ and $\text{TinyBERT}_{4,\text{TS}}$. They retain, respectively, 97.3%, 96.8%, and 96.8% of accuracy. (3) TinyBERT with task-specific KD improves the results

TABLE III
COMPARISON OF THE CONSIDERED COMPRESSION METHODS FOR MULTIEMO ALL DOMAINS AT THE DOCUMENT LEVEL. THE RESULTS ARE AVERAGED ON 5 REPETITIONS.

Method	Parameters	Memory [MB]	Training [min]	Eval [s]	Accuracy	Macro F1	Macro Recall	Macro Precision
$\text{BERT}_{\text{BASE}}$	109M	418	12.7	7.3	86.7	84.8	84.5	85.9
DistilBERT	67M	255 (1.6x)	6.7	3.6 (2.0x)	**87.2**	**85.3**	**85.0**	**86.0**
XtremeDistil	**13M**	**49 (8.6x)**	2.3	1.1 (6.4x)	86.4	85.0	84.8	85.2
$\text{TinyBERT}_{6,\text{TA}}$	67M	255 (1.6x)	6.8	3.7 (2.0x)	85.7	84.8	84.9	84.8
$\text{TinyBERT}_{6,\text{TS}}$	68M	258 (1.6x)	28.7	4.1 (1.8x)	86.2	84.4	83.8	85.1
$\text{TinyBERT}_{4,\text{TA}}$	14M	55 (7.6x)	**2.0**	1.0 (7.1x)	84.3	82.7	82.3	83.2
$\text{TinyBERT}_{4,\text{TS}}$	15M	56 (7.5x)	16.1	1.1 (6.5x)	86.2	84.1	83.2	85.6
BERT-of-Theseus	67M	255 (1.6x)	10.2	3.7 (1.9x)	85.2	83.9	83.8	84.1
ROSITA	14M	55 (7.6x)	55.0	**1.0 (7.5x)**	83.7	81.9	81.8	82.5

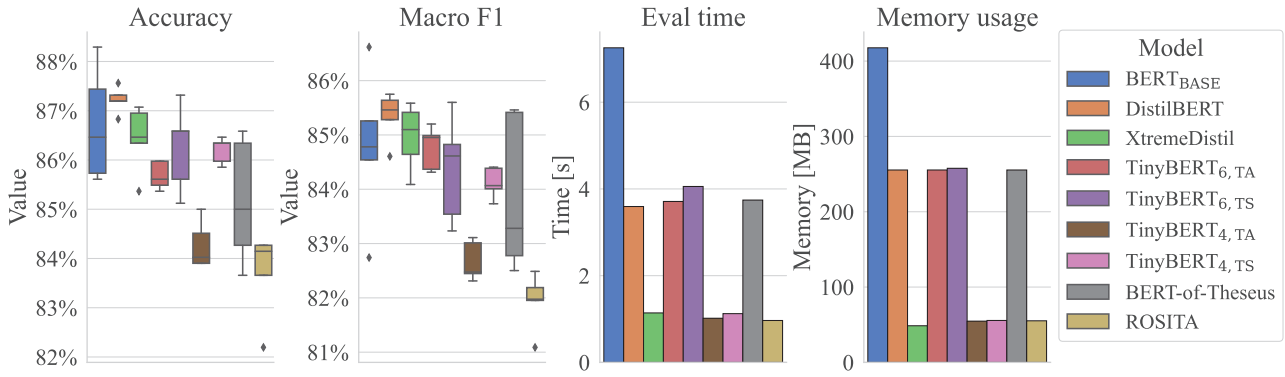

Fig. 2. Comparison between the considered compression methods applied for the document level MultiEmo data for all domains. The results are averaged on five repetitions.

by a slight margin but demands a much longer training procedure (it needs two phases of distillation). (4) DistilBERT offers moderate compression with only a small performance drop. Its results are very similar to the task-agnostic variant of TinyBERT_6 and are only not significantly worse than $\text{TinyBERT}_{6,\text{TS}}$. Task-specific methods are not always needed since more flexible task-agnostic methods can obtain competitive performance. (5) BERT-of-Theseus does not improve the results compared to DistilBERT but is trained longer. It is the worst compression method in that scenario because the compression ratio is not so high, and the performance drop is significant. (6) ROSITA model has comparable size reduction as XtremeDistill and TinyBERT_4 with slightly better results. However, its training process is very long because consists of 3 stages.

Table III and Figure 2 present results of the experiments at the document level. The best results are obtained for DistilBERT_6 which even outperforms the fine-tuned $\text{BERT}_{\text{BASE}}$ model. The same as for the sentence level, the most efficient models are XtremeDistill, $\text{TinyBERT}_{4,\text{TA}}$ and $\text{TinyBERT}_{4,\text{TS}}$ which retain respectively 99.7%, 97.2% and 99.4% of accuracy. In that case, task-specific TinyBERT outperforms task-agnostic TinyBERT more significantly. It shows that task-specific distillation can be more important for tasks with smaller data. BERT-of-Theseus method again is worse than DistilBERT. ROSITA is much worse than the other methods. Its weak performance can be caused by the smaller number of data for the document-level tasks. Additionally, this data was not augmented as it was done in the original paper. It shows that the ROSITA method works very well but only for tasks with a sufficient amount of data.

The mixed domain scenario experiments at both text granularity levels have shown that the best trade-off between efficiency and effectiveness is achieved for TinyBERT_4 and XtremeDistill. Their speedup and size reduction are approximately 7x. ROSITA also works quickly but demands a very long training procedure and a greater amount of data, which not always can be assured. Such great efficiency improvements are not observed for DistilBERT and TinyBERT_6 what makes them a worse option for compression even though their classification quality is better and, in some cases, can even outperform the original model.

A. Domain studies

We do thorough studies on how the considered models work in the following domain settings: (1) single-domain – model trained and evaluated on the same data from a single domain; (2) domain-out – model trained using elements from 3 domains and evaluated on the remaining domain, this variant can verify the classification ability to capture domain-independent

TABLE IV
SINGLE-DOMAIN (SD) RESULTS FOR THE SENTENCE LEVEL (SDS) FOR VARIOUS DOMAINS (HOTELS, MEDICINE, PRODUCTS, SCHOOL).

T	Method	Acc.	F1	SP	AMB	0	SN
SDS-H	$\text{BERT}_{\text{BERT}}$	82.08	77.40	85.87	59.07	77.58	87.09
	DistilBERT	81.50	76.67	85.09	58.65	76.19	86.76
	$\text{TinyBERT}_{6,\text{TA}}$	81.78	77.16	**86.22**	**58.97**	76.62	86.81
	$\text{TinyBERT}_{6,\text{TS}}$	**82.25**	**77.37**	85.75	58.72	**77.41**	**87.61**
	BERT-of-Theseus	80.37	75.27	84.27	55.53	75.31	85.98
	XtremeDistil	**80.66**	**75.87**	84.25	**57.97**	**75.51**	**85.74**
	$\text{TinyBERT}_{4,\text{TA}}$	80.11	75.37	83.55	57.57	74.70	85.65
	$\text{TinyBERT}_{4,\text{TS}}$	80.45	75.25	**84.80**	56.19	74.55	85.47
SDS-M	$\text{BERT}_{\text{BERT}}$	78.64	70.18	81.27	38.45	83.93	77.08
	DistilBERT	78.55	69.79	81.48	36.02	84.08	77.58
	$\text{TinyBERT}_{6,\text{TA}}$	78.01	69.33	81.07	36.14	83.60	76.52
	$\text{TinyBERT}_{6,\text{TS}}$	**79.37**	**71.20**	**82.81**	**39.49**	**84.45**	**78.04**
	BERT-of-Theseus	77.78	68.35	79.83	33.58	83.45	76.55
	XtremeDistil	**77.21**	**68.27**	78.26	**35.44**	**82.96**	**76.40**
	$\text{TinyBERT}_{4,\text{TA}}$	75.94	66.97	76.93	34.01	81.62	75.32
	$\text{TinyBERT}_{4,\text{TS}}$	76.63	67.35	**79.22**	31.85	**82.96**	75.38
SDS-P	$\text{BERT}_{\text{BERT}}$	70.78	63.77	80.18	45.52	49.88	79.52
	DistilBERT	70.16	61.75	75.87	41.37	**49.75**	79.99
	$\text{TinyBERT}_{6,\text{TA}}$	70.00	61.11	77.31	43.27	44.49	79.37
	$\text{TinyBERT}_{6,\text{TS}}$	**71.13**	**63.73**	**77.70**	**49.40**	47.61	**80.22**
	BERT-of-Theseus	68.63	60.33	75.57	38.96	48.50	78.29
	XtremeDistil	**69.84**	**62.42**	**77.91**	40.82	**53.15**	**77.80**
	$\text{TinyBERT}_{4,\text{TA}}$	67.87	58.90	74.54	37.71	46.34	77.04
	$\text{TinyBERT}_{4,\text{TS}}$	68.54	61.42	74.25	**47.39**	46.37	77.68
SDS-S	$\text{BERT}_{\text{BERT}}$	61.90	56.03	67.38	63.35	38.24	55.16
	DistilBERT	62.21	56.41	65.82	**65.21**	40.53	54.07
	$\text{TinyBERT}_{6,\text{TA}}$	61.74	57.59	68.14	62.16	**46.87**	53.20
	$\text{TinyBERT}_{6,\text{TS}}$	**63.72**	**57.90**	**70.34**	64.92	39.52	**56.81**
	BERT-of-Theseus	57.00	49.34	63.90	60.31	38.25	34.90
	XtremeDistil	60.16	52.55	65.16	63.81	43.65	37.59
	$\text{TinyBERT}_{4,\text{TA}}$	**62.21**	**57.55**	**65.18**	**65.02**	**44.41**	**55.58**
	$\text{TinyBERT}_{4,\text{TS}}$	57.55	53.18	61.04	60.64	41.09	49.94

TABLE V
SINGLE-DOMAIN (SD) RESULTS FOR THE WHOLE TEXT LEVEL (SDT) FOR VARIOUS DOMAINS (HOTELS, MEDICINE, PRODUCTS, SCHOOL).

T	Method	Acc.	F1	SP	AMB	0	SN
SDT-H	$\text{BERT}_{\text{BERT}}$	85.06	84.48	88.47	62.94	96.57	89.94
	DistilBERT	**86.84**	**87.01**	88.95	**70.10**	97.85	**91.13**
	$\text{TinyBERT}_{6,\text{TA}}$	85.82	86.14	88.67	67.11	**98.55**	90.23
	$\text{TinyBERT}_{6,\text{TS}}$	84.61	84.56	88.17	62.75	97.80	89.51
	BERT-of-Theseus	84.00	83.86	**89.09**	62.22	96.05	88.09
	XtremeDistil	**86.08**	**85.72**	**88.77**	**66.75**	96.55	90.81
	$\text{TinyBERT}_{4,\text{TA}}$	85.01	84.59	86.19	64.01	97.05	**91.10**
	$\text{TinyBERT}_{4,\text{TS}}$	84.15	84.28	87.32	63.08	**97.59**	89.13
SDT-M	$\text{BERT}_{\text{BERT}}$	90.40	81.23	92.03	42.82	98.40	91.66
	DistilBERT	**88.93**	79.49	90.75	39.05	97.72	**90.45**
	$\text{TinyBERT}_{6,\text{TA}}$	88.32	**79.82**	**90.76**	**41.23**	**98.26**	89.05
	$\text{TinyBERT}_{6,\text{TS}}$	79.08	70.22	81.31	32.35	96.69	70.52
	BERT-of-Theseus	86.36	77.93	87.61	38.74	97.56	87.80
	XtremeDistil	**89.42**	77.42	90.58	29.91	**98.39**	**90.80**
	$\text{TinyBERT}_{4,\text{TA}}$	84.40	74.03	84.52	28.76	95.88	86.95
	$\text{TinyBERT}_{4,\text{TS}}$	86.97	**80.11**	**90.93**	**44.49**	97.26	87.76
SDT-P	$\text{BERT}_{\text{BERT}}$	75.00	43.36	30.00	57.97	00.00	85.47
	DistilBERT	**79.58**	**46.96**	**33.33**	**67.22**	00.00	**87.29**
	$\text{TinyBERT}_{6,\text{TA}}$	74.58	41.69	28.00	54.22	00.00	84.53
	$\text{TinyBERT}_{6,\text{TS}}$	75.42	34.14	00.00	51.69	00.00	84.86
	BERT-of-Theseus	68.33	34.52	8.00	48.95	00.00	81.15
	XtremeDistil	**80.83**	**44.19**	**20.00**	**67.53**	00.00	**89.22**
	$\text{TinyBERT}_{4,\text{TA}}$	72.08	28.55	00.00	31.18	00.00	83.03
	$\text{TinyBERT}_{4,\text{TS}}$	80.00	33.47	00.00	44.87	00.00	89.00
SDT-S	$\text{BERT}_{\text{BERT}}$	75.20	45.08	83.46	68.86	00.00	28.00
	DistilBERT	**77.60**	**47.47**	**85.01**	**72.88**	00.00	**32.00**
	$\text{TinyBERT}_{6,\text{TA}}$	74.00	39.65	82.76	69.16	00.00	6.67
	$\text{TinyBERT}_{6,\text{TS}}$	68.00	32.14	79.42	49.14	00.00	00.00
	BERT-of-Theseus	68.80	32.51	78.56	51.49	00.00	00.00
	XtremeDistil	**76.00**	**38.72**	**83.98**	**70.89**	00.00	00.00
	$\text{TinyBERT}_{4,\text{TA}}$	72.80	36.58	82.17	64.16	00.00	00.00
	$\text{TinyBERT}_{4,\text{TS}}$	70.00	35.39	79.96	61.59	00.00	00.00

sentiment features. These experiments were performed for all models apart from ROSITA due to the long training time.

Table IV and Table V present single-domain results for the sentence and the document level, respectively. Accuracy and macro F1-score are presented along with F1-score for each sentiment class. Shortened names of the methods are used. DistilBERT, TinyBERT_6, and BERT-of-Theseus are grouped as larger models compared to the most efficient XtremeDistil and TinyBERT_4 methods. For both text granularities, the results are better for domains with more data, i.e., hotels and medicine. Performance of the compressed models is comparable with $\text{BERT}_{\text{BASE}}$; in many cases, it is even better, especially for domains with smaller amounts of data. For the sentence level, $\text{TinyBERT}_{6,\text{TS}}$ obtains the highest results, but for the text level, DistilBERT is better. It shows that when there is a sufficient amount of training examples, task-specific methods give better results than task-agnostic ones, but it can be the opposite when there is little data.

For the sentence level among the most efficient methods, the best results are obtained for XtremeDistil in most cases (only for the school domain $\text{TinyBERT}_{4,\text{TA}}$ is better). Its scores are approximately 2 percentage points worse than the best ones. At the text level, XtremeDistil outperforms other methods within the most efficient method group. Moreover, its results demonstrate the same quality level as the larger methods, which is not observed at the sentence level. The best compressing method outperforms the original model in 7 out of 8 cases. It suggests that a smaller, more appropriate model can be better for a simple task, especially when the dataset is not very large. The higher number of parameters can cause that model to overfit, which is easier to avoid in the case of smaller models.

Looking at the particular sentiment classes, the best results are obtained for positive and negative examples for all domains. It means that more expressive sentiment is easier to be recognized by the models, which can be confirmed by the results for neutral class, which are generally slightly worse (apart from the hotels and medicine domains for the text level). The worst performance is for an ambivalent class. The models have problems detecting that class because it is not easy to define, as it consists of ambiguous texts that were not easy to annotate even for people.

Domain-out results are presented in Table VI and Table VII. For all domains, the results are worse than in the corresponding single-domain scenario. In many cases, the results for the compressed methods are better than for $\text{BERT}_{\text{BASE}}$. Large size causes the model to be prone to overfitting to trained data. Thus it loses its generalization abilities for data from other domains. For the compressed models, the adversarial phenomenon is lesser; e.g., DistilBERT, XtremeDistil, or $\text{TinyBERT}_{6,\text{TA}}$ outperforms BERT in many cases, in particular for the text level. The results demonstrate that task-agnostic

TABLE VI
DOMAIN-OUT (DO) RESULTS FOR THE SENTENCE LEVEL (DOS) FOR VARIOUS *out* DOMAINS: (**H**OTELS, **M**EDICINE, **P**RODUCTS, **S**CHOOL).

T	Method	Acc.	F1	SP	AMB	0	SN
DOS-H	$\text{BERT}_{\text{BERT}}$	74.28	69.18	79.95	50.94	64.99	80.84
	DistilBERT	71.97	66.76	77.87	47.77	63.10	78.32
	$\text{TinyBERT}_{6,\text{TA}}$	72.59	67.61	78.78	49.09	63.80	78.75
	$\text{TinyBERT}_{6,\text{TS}}$	**75.58**	**70.05**	**79.22**	**53.07**	**65.95**	**81.97**
	BERT-of-Theseus	70.36	65.39	76.66	47.27	60.45	77.17
	XtremeDistil	**74.05**	**68.53**	**78.21**	**49.74**	**65.40**	**80.75**
	$\text{TinyBERT}_{4,\text{TA}}$	71.70	66.74	76.02	49.15	63.37	78.41
	$\text{TinyBERT}_{4,\text{TS}}$	71.76	65.29	75.26	45.28	61.42	79.20
DOS-M	$\text{BERT}_{\text{BERT}}$	66.78	60.98	71.30	31.96	70.69	69.98
	DistilBERT	65.88	59.65	70.20	31.08	67.97	69.34
	$\text{TinyBERT}_{6,\text{TA}}$	65.63	**60.17**	69.81	**32.66**	68.31	69.91
	$\text{TinyBERT}_{6,\text{TS}}$	**68.05**	59.67	**70.77**	25.98	**71.25**	**70.66**
	BERT-of-Theseus	64.30	58.24	66.50	29.37	68.91	68.19
	XtremeDistil	**66.25**	**60.14**	**68.67**	**33.61**	**69.14**	**69.15**
	$\text{TinyBERT}_{4,\text{TA}}$	63.82	58.35	66.74	33.36	65.76	67.52
	$\text{TinyBERT}_{4,\text{TS}}$	65.02	56.71	67.45	23.91	67.57	67.92
DOS-P	$\text{BERT}_{\text{BERT}}$	62.18	55.35	70.92	39.85	37.22	73.40
	DistilBERT	61.51	53.88	**70.57**	36.50	35.17	**73.27**
	$\text{TinyBERT}_{6,\text{TA}}$	60.92	54.64	70.41	39.72	**36.76**	71.65
	$\text{TinyBERT}_{6,\text{TS}}$	**61.67**	**54.77**	70.00	**41.14**	35.12	72.83
	BERT-of-Theseus	60.59	53.52	68.74	39.94	33.05	72.36
	XtremeDistil	**63.91**	**56.66**	**74.94**	38.66	**39.41**	**73.63**
	$\text{TinyBERT}_{4,\text{TA}}$	59.35	52.25	69.87	35.77	33.52	69.84
	$\text{TinyBERT}_{4,\text{TS}}$	58.52	51.95	64.06	**41.29**	31.90	70.54
DOS-S	$\text{BERT}_{\text{BERT}}$	50.04	46.17	58.36	49.96	27.18	49.18
	DistilBERT	50.59	46.70	**63.42**	43.99	**29.82**	49.56
	$\text{TinyBERT}_{6,\text{TA}}$	**53.20**	**48.58**	62.69	52.46	27.80	**51.39**
	$\text{TinyBERT}_{6,\text{TS}}$	52.73	48.56	58.59	**55.14**	29.62	50.89
	BERT-of-Theseus	49.09	44.31	59.22	49.61	20.80	47.61
	XtremeDistil	**52.65**	**47.54**	**62.77**	**50.35**	**24.04**	**53.01**
	$\text{TinyBERT}_{4,\text{TA}}$	47.75	42.75	59.80	45.39	16.90	48.89
	$\text{TinyBERT}_{4,\text{TS}}$	46.96	42.74	57.68	46.11	23.37	43.79

TABLE VII
DOMAIN-OUT (DO) RESULTS FOR THE WHOLE TEXT LEVEL (DOT) FOR VARIOUS *out* DOMAINS: (**H**OTELS, **M**EDICINE, **P**RODUCTS, **S**CHOOL).

T	Method	Acc.	F1	SP	AMB	0	SN
DOT-H	$\text{BERT}_{\text{BERT}}$	80.20	78.12	84.21	50.64	91.01	86.60
	DistilBERT	**81.37**	**80.16**	**86.26**	57.19	90.40	**86.80**
	$\text{TinyBERT}_{6,\text{TA}}$	80.76	80.09	85.68	**57.81**	**90.50**	86.36
	$\text{TinyBERT}_{6,\text{TS}}$	74.53	70.63	81.51	39.77	78.27	82.98
	BERT-of-Theseus	76.66	74.55	83.56	44.66	86.86	83.09
	XtremeDistil	78.94	77.00	82.99	47.48	**90.90**	86.62
	$\text{TinyBERT}_{4,\text{TA}}$	**79.49**	**77.87**	81.39	**55.72**	87.00	**87.38**
	$\text{TinyBERT}_{4,\text{TS}}$	76.86	73.11	**83.58**	42.41	82.08	84.36
DOT-M	$\text{BERT}_{\text{BERT}}$	82.57	74.52	85.66	31.61	94.33	86.46
	DistilBERT	**81.35**	**72.83**	**83.94**	32.05	**89.57**	**85.77**
	$\text{TinyBERT}_{6,\text{TA}}$	77.61	71.46	79.07	35.46	88.80	82.50
	$\text{TinyBERT}_{6,\text{TS}}$	69.24	59.18	61.45	26.65	70.03	78.59
	BERT-of-Theseus	77.74	71.20	83.23	**33.44**	87.00	81.14
	XtremeDistil	**83.24**	**75.41**	**85.95**	**35.82**	**93.57**	**86.30**
	$\text{TinyBERT}_{4,\text{TA}}$	76.45	69.17	76.49	27.90	89.21	83.07
	$\text{TinyBERT}_{4,\text{TS}}$	75.05	68.22	72.33	27.90	88.47	84.19
DOT-P	$\text{BERT}_{\text{BERT}}$	81.67	68.21	63.33	67.21	53.33	88.95
	DistilBERT	**82.08**	**66.92**	54.67	**70.38**	53.33	**89.32**
	$\text{TinyBERT}_{6,\text{TA}}$	69.17	61.05	59.33	50.55	**54.67**	79.65
	$\text{TinyBERT}_{6,\text{TS}}$	67.92	52.67	67.43	50.46	11.11	81.67
	BERT-of-Theseus	70.00	49.20	62.86	53.40	00.00	80.54
	XtremeDistil	**80.42**	**64.77**	64.10	**66.00**	**40.00**	**88.98**
	$\text{TinyBERT}_{4,\text{TA}}$	71.67	50.44	57.43	61.63	00.00	82.68
	$\text{TinyBERT}_{4,\text{TS}}$	77.50	53.96	**72.00**	56.81	00.00	87.05
DOT-S	$\text{BERT}_{\text{BERT}}$	68.40	48.56	81.64	59.44	00.00	53.17
	DistilBERT	68.80	48.40	83.36	61.36	00.00	48.89
	$\text{TinyBERT}_{6,\text{TA}}$	66.40	46.51	79.23	56.86	00.00	49.95
	$\text{TinyBERT}_{6,\text{TS}}$	62.00	42.96	77.04	48.57	00.00	46.24
	BERT-of-Theseus	**73.20**	**53.02**	**86.95**	**67.06**	00.00	**58.06**
	XtremeDistil	65.60	42.89	85.32	51.35	00.00	34.88
	$\text{TinyBERT}_{4,\text{TA}}$	**69.60**	**52.70**	**82.84**	**68.07**	00.00	**59.90**
	$\text{TinyBERT}_{4,\text{TS}}$	59.60	42.21	77.88	51.99	00.00	38.98

methods are better in that scenario. Task-specific models can give worse results because they lose generalizing abilities in compensation for better performing on close-to-train-data examples.

The observations for the specific sentiment categories are consistent with those noted for the single-domain scenario. The polarized categories have better results than the neutral and ambivalent categories. The cross-domain approach allows the models to better detect positive texts in the product domain and negative texts in the school domain, which was impossible for the single domain scenario. It shows that the domain-out approach can be beneficial in some specific cases. But this is not always true; for example, for TinyBERT_4 models for neutral class in the product domain.

The domain experiments have shown that task-specific methods perform better for single-domain and domain-out scenarios, but only when a large dataset is necessary. On the other hand, these methods work worse when the dataset is limited. In those cases, task-specific distillation is not so effective, and it is better to rely on the pre-distilled model, which is fine-tuned for a concrete task. In the context of the domain-out experiment, another factor is overfitting; it is possible that due to the distillation of other domains with insufficient data, the model loses generalizing abilities. Hence its performance on a left-out domain is worse.

The analyses of the results reveal that the compressed models can outperform the original model when working with limited data. The lesser number of parameters makes such models less prone to overfit. It is especially relevant for cross-domain cases. Among the most efficient models, XtremeDistil outperforms both TinyBERT_4 models in most cases. It is also competitive with the larger models, especially at the text level.

VI. CONCLUSIONS

In this work, we compared transformer compression methods focusing on the MultiEmo sentiment analysis task. The experiments showed that the performance of the compressed method is slightly worse, but much better efficiency is gained. XtremeDisitl method is the best trade-off between efficiency and quality because it offers a 6.4x speed-up and 8.6x compression rate with a minimal performance drop.

The compressed method, especially task-agnostic methods like XtremeDistil, can outperform the original size model mainly when data are limited. Smaller models are not so prone to overfitting, so their application can be treated as a kind of regularization compared to larger models. That aspect can be even more important in cross-domain situations. Similarly, task-agnostic methods have better generalization ability than task-specific methods when the amount of data is limited, as was shown in the domain-out experiments. However, when high performance is required for a concrete task, task-specific methods should still be considered, especially when the size of a dataset is considerable.

The high performance of the XtremeDistil method with such a large compression ratio can arise from applying the ELECTRA model as a teacher, which is different from the other methods where standard $\text{BERT}_{\text{BASE}}$ is used. It is also initialized with the MiniLM model, which is pre-distilled from $\text{BERT}_{\text{BASE}}$. It shows that a better teacher and starting point can help obtain better results. Future ablation studies should be conducted to check whether it is the main factor causing its superiority over the other methods. Additionally, TinyBERT, for which good results are obtained, initially uses data augmentation. In future work, it is worth verifying how it improves the quality of the ROSITA method.

REFERENCES

[1] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *NAACL*, 2018.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2019.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[4] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," *ArXiv*, vol. abs/2003.10555, 2020.

[5] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *ArXiv*, vol. abs/1706.03762, 2017.

[6] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," *ArXiv*, vol. abs/1804.07461, 2018.

[7] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, "Affective computing and sentiment analysis," in *A practical guide to sentiment analysis*. Springer, 2017, pp. 1–10.

[8] E. Cambria, Q. Liu, S. Decherchi, F. Xing, and K. Kwok, "Senticnet 7: a commonsense-based neurosymbolic ai framework for explainable sentiment analysis," *Proceedings of LREC 2022*, 2022.

[9] J. Ramteke, S. Shah, D. Godhia, and A. Shaikh, "Election result prediction using twitter sentiment analysis," 2016.

[10] S. Vairavasundaram, L. Ravi, M. Abejith, S. Umasankar, and A. Umamakeswari, "Sentiment analysis of tweets for estimating criticality and security of events," *J. Organ. End User Comput.*, vol. 29, pp. 51–71, 2017.

[11] M.-Y. Day and Y.-D. Lin, "Deep learning for sentiment analysis on google play consumer review," *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, 2017.

[12] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment analysis for e-commerce product reviews in chinese based on sentiment lexicon and deep learning," *IEEE Access*, 2020.

[13] K. Mishev, A. Gjorgjevikj, I. Vodenska, L. Chitkushev, and D. Trajanov, "Evaluation of sentiment analysis in finance: From lexicons to transformers," *IEEE Access*, 2020.

[14] Y. Gong, L. Liu, M. Yang, and L. D. Bourdev, "Compressing deep convolutional networks using vector quantization," *ArXiv*, vol. abs/1412.6115, 2014.

[15] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," 2016.

[16] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv*, vol. abs/1503.02531, 2015.

[17] O. Zafrir, G. Boudoukh, P. Izsak, and M. Wasserblat, "Q8BERT: Quantized 8Bit BERT," *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*, pp. 36–39, 2019.

[18] W. Zhang, L. Hou, Y. Yin, L. Shang, X. Chen, X. Jiang, and Q. Liu, "TernaryBERT: Distillation-aware ultra-low bit BERT," 2020.

[19] H. Bai, W. Zhang, L. Hou, L. Shang, J. Jin, X. Jiang, Q. Liu, M. Lyu, and I. King, "BinaryBERT: Pushing the limit of BERT quantization," 2021.

[20] M. A. Gordon, K. Duh, and N. Andrews, "Compressing BERT: Studying the effects of weight pruning on transfer learning," 2020.

[21] H. Sajjad, F. Dalvi, N. Durrani, and P. Nakov, "Poor man's BERT: Smaller and faster transformer models," *ArXiv*, vol. abs/2004.03844, 2020.

[22] C. Xu, W. Zhou, T. Ge, F. Wei, and M. Zhou, "BERT-of-Theseus: Compressing bert by progressive module replacing," 2020.

[23] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient knowledge distillation for BERT model compression," in *EMNLP*, 2019.

[24] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," 2020.

[25] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," 2020.

[26] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," 2020.

[27] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "MobileBERT: a compact task-agnostic BERT for resource-limited devices," *ArXiv*, vol. abs/2004.02984, 2020.

[28] Y. Mao, Y. Wang, C. Wu, C. Zhang, Y. Wang, Y. Yang, Q. Zhang, Y. Tong, and J. Bai, "LadaBERT: Lightweight adaptation of BERT through hybrid model compression," in *COLING*, 2020.

[29] L. Hou, Z. Huang, L. Shang, X. Jiang, X. Chen, and Q. Liu, "DynaBERT: Dynamic BERT with adaptive width and depth," 2020.

[30] S. Mukherjee, A. H. Awadallah, and J. Gao, "XtremeDistilTransformers: Task transfer for task-agnostic distillation," 2021.

[31] Y. Liu, Z. Lin, and F. Yuan, "ROSITA: Refined BERT compression with integrated techniques," *ArXiv*, vol. abs/2103.11367, 2021.

[32] J. Kocoń, P. Miłkowski, and K. Kanclerz, *MultiEmo: Multilingual, Multilevel, Multidomain Sentiment Analysis Corpus of Consumer Reviews*, 06 2021, pp. 297–312.

[33] C. Bucila, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *KDD '06*, 2006.

[34] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2015.

[35] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," in *International Conference on Learning Representations*, 2018.

[36] S. I. Mirzadeh, M. Farajtabar, A. Li, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher," 2019.

[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[38] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[39] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2017.

[40] W. Zhang and S. Skiena, "Trading strategies to exploit blog and news sentiment." 2010.

[41] W. Liu, P. Zhou, Z. Zhao, Z. Wang, H. Deng, and Q. Ju, "FastBERT: a self-distilling BERT with adaptive inference time," 2020.

[42] S. Geng, P. Gao, Z. Fu, and Y. Zhang, "RomeBERT: Robust training of multi-exit BERT," 2021.

[43] P. Ganesh, Y. Chen, X. Lou, M. A. Khan, Y. Yang, D. Chen, M. Winslett, H. Sajjad, and P. Nakov, "Compressing large-scale transformer-based models: A case study on BERT," *Transactions of the Association for Computational Linguistics*, 2021.

[44] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *EMNLP*, 2016.

[45] J. Kocoń, P. Miłkowski, and M. Zaśko-Zielińska, "Multi-level sentiment analysis of polemo 2.0: Extended corpus of multi-domain consumer reviews," in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 2019.

[46] K. Kanclerz, P. Miłkowski, and J. Kocoń, "Cross-lingual deep neural transfer learning in sentiment analysis," *Procedia Computer Science*, vol. 176, pp. 128–137, 2020.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.