Extremely Small BERT Models from Mixed-Vocabulary Training

Sanqiang Zhao*
University of Pittsburgh, PA
sanqiang.zhao@pitt.edu

Raghav Gupta*
Google Research, Mountain View, CA
raghavgupta@google.com

Yang Song
Kuaishou Technology, Beijing, China
yangsong@kuaishou.com

Denny Zhou
Google Brain, Mountain View, CA
dennyzhou@google.com

Abstract

Pretrained language models like BERT have achieved good results on NLP tasks, but are impractical on resource-limited devices due to memory footprint. A large fraction of this footprint comes from the input embeddings with large input vocabulary and embedding dimensions. Existing knowledge distillation methods used for model compression cannot be directly applied to train student models with reduced vocabulary sizes. To this end, we propose a distillation method to align the teacher and student embeddings via mixed-vocabulary training. Our method compresses BERT$_{\text{LARGE}}$ to a task-agnostic model with smaller vocabulary and hidden dimensions, which is an order of magnitude smaller than other distilled BERT models and offers a better size-accuracy trade-off on language understanding benchmarks as well as a practical dialogue task.

1 Introduction

Recently, pre-trained context-aware language models like ELMo (Peters et al., 2018), GPT (Radford et al., 2019), BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019) have outperformed traditional word embedding models like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), and achieved strong results on a number of language understanding tasks. However, these models are typically too huge to host on mobile/edge devices, especially for real-time inference. Recent work has explored, inter alia, knowledge distillation (Ba and Caruana, 2014; Hinton et al., 2015) to train small-footprint student models by implicit transfer of knowledge from a teacher model.

Most distillation methods, however, need the student and teacher output spaces to be aligned. This complicates task-agnostic distillation of BERT to smaller-vocabulary student BERT models since the input vocabulary is also the output space for the masked language modeling (MLM) task used in BERT. This in turn limits these distillation methods' ability to compress the input embedding matrix, that makes up a major proportion of model parameters e.g. the ~30K input WordPiece embeddings of the BERT$_{\text{BASE}}$ model make up over 21% of the model size. This proportion is even higher for most distilled BERT models, owing to these distilled models typically having fewer layers than their teacher BERT counterparts.

We present a task and model-agnostic distillation approach for training small, reduced-vocabulary BERT models running into a few megabytes. In our setup, the teacher and student models have incompatible vocabularies and tokenizations for the same sequence. We therefore align the student and teacher WordPiece embeddings by training the teacher on the MLM task with a mix of teacher-tokenized and student-tokenized words in a sequence, and then using these student embeddings to train smaller student models. Using our method, we train compact 6 and 12-layer reduced-vocabulary student models which achieve competitive performance in addition to high compression for benchmark datasets as well as a real-world application in language understanding for dialogue.

2 Related Work

Work in NLP model compression falls broadly into four classes: matrix approximation, weight quantization, pruning/sharing, and knowledge distillation.

The former two seek to map model parameters to low-rank approximations (Tulloch and Jia, 2017) and lower-precision integers/floats (Chen et al., 2015; Zhou et al., 2018; Shen et al., 2019) respectively. In contrast, pruning aims to remove/share redundant model weights (Li et al., 2016; Lan et al., 2019). More recently, dropout (Srivastava et al., 2014) has been used to cut inference latency by

Asterisk (*) denotes equal contribution. Research conducted when all authors were at Google.

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 2753–2759
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

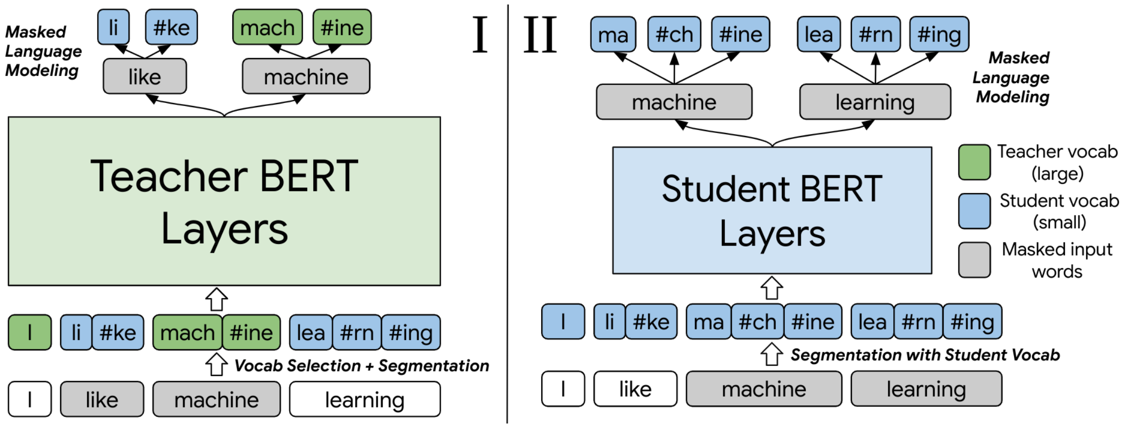

Figure 1: Depiction of our mixed-vocabulary training approach. (Left) Stage I involving retrained teacher BERT with default config (e.g., 30K vocabulary, 768 hidden dim) and mixed-vocabulary input. (Right) Stage II involving student model with smaller vocabulary (5K) and hidden dims (e.g., 256) and embeddings initialized from stage I.

early exit (Fan et al., 2019; Xin et al., 2020).

Knowledge distillation focuses on implicit transfer of knowledge as soft teacher predictions (Tang et al., 2019), attention distributions (Zagoruyko and Komodakis, 2016) and intermediate outputs (Romero et al., 2014). Approaches close to our work rely on similar methods (Sanh et al., 2019; Sun et al., 2019), while others involve combinations of layer-wise transfer (Sun et al., 2020), task-specific distillation (Jiao et al., 2019), architecture search (Chen et al., 2020) and layer dropout (Xu et al., 2020); many of these are specific to the transformer layer (Vaswani et al., 2017).

Another highly relevant line of work focuses on reducing the size of the embedding matrix, either via factorization (Shu and Nakayama, 2018; Lan et al., 2019) or vocabulary selection/pruning (Provilkov et al., 2019; Chen et al., 2019b).

3 Proposed Approach

Here, we discuss our rationale behind reducing the student vocabulary size and its challenges, followed by our *mixed-vocabulary* distillation approach.

3.1 Student Vocabulary

WordPiece (WP) tokens (Wu et al., 2016) are subword units obtained by applying greedy segmentation to a training corpus. Given such a corpus and a number of desired tokens D, a WordPiece vocabulary is generated by selecting D subword tokens such that the resulting corpus is minimal in the number of WordPiece when segmented according to the chosen WordPiece model. The greedy algorithm for this optimization problem is described in more detail in Sennrich et al. (2016). Most published BERT models use a vocabulary of 30522 WordPieces, obtained by running the above algorithm on the Wikipedia and BooksCorpus (Zhu et al., 2015) corpora with a desired vocabulary size D of 30000.

For our student model, we chose a target vocabulary size D of 5000 WordPiece tokens. Using the same WordPiece vocabulary generation algorithm and corpus as above, we obtain a 4928-WordPiece vocabulary for the student model. This student vocabulary includes all ASCII characters as separate tokens, ensuring no out-of-vocabulary words upon tokenization with this vocabulary. Additionally, the 30K teacher BERT vocabulary includes 93.9% of the WP tokens in this 5K student vocabulary but does not subsume it. We explore other strategies to obtain a small student vocabulary in Section 6.

For task-agnostic student models, we reuse BERT's masked language modeling (MLM) task: words in context are randomly masked and predicted given the context via softmax over the model's WP vocabulary. Thus, the output spaces for our teacher (30K) and student (5K) models are unaligned. This, coupled with both vocabularies tokenizing the same words differently, means existing distillation methods do not apply to our setting.

3.2 Mixed-vocabulary training

We propose a two-stage approach for implicit transfer of knowledge to the student via the student embeddings, as described below.

Stage I (Student Embedding Initialization): We first train the student embeddings with the teacher model initialized from BERT$_{\text{LARGE}}$. For a given input sequence, we mix the vocabularies by randomly selecting (with probability p_{SV}, a hyperparameter) words from the sequence to segment using the student vocabulary, with the other words segmented using the teacher vocabulary. As in Figure 1 on the left, for input [‘*I*’, ‘*like*’, ‘*machine*’, ‘*learning*’], the words *‘like’* and *‘learning’* are segmented using the student vocabulary (in blue), with the others using the teacher vocabulary (in green). Similar to Lample and Conneau (2019), this step seeks to align the student and teacher embeddings for the same tokens: the model learns to predict student tokens using context which is segmented using the teacher vocabulary, and vice versa.

Note that since the student embeddings are set to a lower dimension than the teacher embeddings, as they are meant to be used in the smaller student model, we project the student embeddings up to the teacher embedding dimension using a trainable affine layer before these are input to the teacher BERT. We choose to keep the two embedding matrices separate despite the high token overlap: this is partly to keep our approach robust to lower vocabulary overlap settings, and partly due to empirical considerations described in Section 6.

Let θ_s/eb_s and θ_t/eb_t denote the transformer layer and embedding weights for the student and teacher models respectively. The loss defined in Equation 1 is the MLM cross entropy summed over masked positions M_t in the teacher input. y_i and c_i denote the predicted and true tokens at position i respectively and can belong to either vocabulary. $v_i{\in}\{s{,}t\}$ denotes the vocabulary used to segment this token. Separate softmax layers P_{v_i} are used for token prediction, one for each vocabulary, depending on the segmenting vocabulary v_i for token i. All teacher parameters (θ_t, eb_t) and student embeddings (eb_s) are updated in this step.

$$L_{s_1} = -\textstyle\sum_{i\in M_t}(\log P_{v_i}(y_i{=}c_i|\theta_t, eb_s, eb_t)) \quad (1)$$

Stage II (Student Model Layers): With student embeddings initialized in stage I, we now train the student model normally i.e., using only the student vocabulary and discarding the teacher model. Equation 2 shows the student MLM loss where M_s is the set of positions masked in the student input. All student model parameters (θ_s, eb_s) are updated.

$$L_{s_2} = -\textstyle\sum_{i\in M_s} \log P_s(y_i{=}c_i|\theta_s, eb_s)) \quad (2)$$

4 Experiments

For evaluation, we finetune the student model just as one would finetune the original BERT model i.e., without using the teacher model or any task-specific distillation. We describe our experiments below, with dataset details left to the appendix.

4.1 Evaluation Tasks and Datasets

We fine-tune and evaluate the distilled student models on two classes of language understanding tasks:

GLUE benchmark (Wang et al., 2019)**:** We pick three classification tasks from GLUE:

- MRPC: Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005), a 2-way sentence pair classification task with 3.7K train instances.
- MNLI: Multi-Genre Natural Language Inference (Williams et al., 2018), a 3-way sentence pair classification task with 393K training instances.
- SST-2: Stanford Sentiment Treebank (Socher et al., 2013), a 2-way sentence classification task with 67K training instances.

Spoken Language Understanding: Since we are also keen on edge device applications, we also evaluate on spoken language understanding, a practical task in dialogue systems. We use the SNIPS dataset (Coucke et al., 2018) of ∼14K virtual assistant queries, each comprising one of 7 intents and values for one or more of the 39 pre-defined slots. The intent detection and slot filling subtasks are modeled respectively as 7-way sentence classification and sequence tagging with IOB slot labels.

4.2 Models and Baselines

For GLUE, we train student models with 6 and 12 layers, 4 attention heads, and embedding/hidden dimensions fixed to 256, each using a compact 5K-WP vocabulary. We also evaluate baselines without knowledge distillation (*NoKD*), parameterized identically to the distilled student models (incl. the 5K vocabulary), trained on the MLM teacher objective from scratch. We also compare our models on GLUE with the following approaches:

- *DistilBERT* (Sanh et al., 2019) distill BERT$_{\text{BASE}}$ to 4/6 layers by aligning teacher predictions,
- *Patient KD - PKD* (Sun et al., 2019) align hidden states to distill BERT$_{\text{BASE}}$ to 3/6 layers,
- *BERT-of-Theseus* (Xu et al., 2020) use a layer dropout method to distill BERT$_{\text{BASE}}$ to 6 layers,
- *TinyBERT* (Jiao et al., 2019) apply task specific distillation to BERT$_{\text{BASE}}$ and align teacher outputs, hidden states as well as embeddings, and
- *MobileBERT* (Sun et al., 2020) combine layer-wise transfer, architecture search and bottleneck

Model	#Params	MRPC	MNLI-m/mm	SST-2	Average
		(F1/Acc)	(Acc)	(Acc)	(F1/Acc)
BERT$_{\text{BASE}}$ (Devlin et al., 2018)	109M	88.9/-	84.6/83.4	93.5	89.0
BERT$_{\text{LARGE}}$ (Devlin et al., 2018)	340M	89.3/-	86.7/85.9	94.9	90.3
PKD$_6$ (Sun et al., 2019)	67.0M	85.0/79.9	81.5/81.0	92.0	86.2
PKD$_3$ (Sun et al., 2019)	45.7M	80.7/72.5	76.7/76.3	87.5	81.6
DistilBERT$_4$ (Sanh et al., 2019)	52.2M	82.4/-	78.9/78.0	91.4	84.2
MobileBERT (Sun et al., 2020)	25.3M	88.8/84.5	83.3/82.6	92.8	88.3
TinyBERT$_4$ (Jiao et al., 2019)	14.5M	82.0*/ -	76.6/77.2*	-	-
TinyBERT$_4$† (Jiao et al., 2019)	14.5M	86.4/-	82.5/81.8	92.6	87.2
BERT-of-Theseus$_6$† (Xu et al., 2020)	66M	87.6/83.2	82.4/82.1	92.2	87.4
NoKD Baseline, L-6, H-256	6.2M	81.2/74.1	76.9/76.1	87.0	81.7
Mixed-vocab distilled (ours), L-6, H-256		84.9/79.3	79.0/78.6	89.1	84.3
NoKD Baseline, L-12, H-256	10.9M	85.1/79.8	79.1/79.0	89.4	84.5
Mixed-vocab distilled (ours), L-12, H-256		87.2/82.6	80.7/80.5	90.6	86.2

* denotes metrics on the development set

† denotes results with task-specific distillation

Table 1: Test set accuracy of distilled models, teacher model and baselines on the GLUE test sets. MNLI-m and MNLI-mm refer to the genre-matched and mismatched test sets. All models other than *NoKD* and our distilled models use a 30K-WordPiece vocabulary. The average uses F1 score for MRPC, accuracy for MNLI-m/SST-2.

structures for an optimized student model.

For SNIPS, we shift our focus to smaller, low-latency models for on-device use cases. Here, we train student models with 6 layers and embedding/hidden dimensions $\in \{96, 192, 256\}$. The smaller models here may not be competitive on GLUE but are adequate for practical tasks such as spoken LU. We compare with two strong baselines:

- BERT$_{\text{BASE}}$ (Chen et al., 2019a) with intent and IOB slot tags predicted using the `[CLS]` and the first WP tokens of each word respectively, and
- StackProp (Qin et al., 2019), which uses a series of smaller recurrent and self-attentive encoders.

4.3 Training Details

Distillation: For all our models, we train the teacher model with mixed-vocabulary inputs (stage I) for 500K steps, followed by 300K steps of training just the student model (stage II). We utilize the same corpora as the teacher model i.e. BooksCorpus (Zhu et al., 2015) and English Wikipedia.

For both stages, up to 20 input tokens were masked for MLM. In stage I, up to 10 of these masked tokens were tokenized using the teacher vocabulary, the rest using the student vocabulary.

We optimize the loss using LAMB (You et al., 2019) with a max learning rate of .00125, linear warmup for the first 10% of steps, batch size of 2048 and sequence length of 128. Distillation was done on Cloud TPUs in a 8x8 pod configuration. p_{SV}, the probability of segmenting a Stage I input word using the student vocabulary, is set to 0.5.

Finetuning: For all downstream task evaluations on GLUE, we finetune for 10 epochs using LAMB with a learning rate of 0.0001 and batch size of 64. For all experiments on SNIPS, we use ADAM with a learning rate of 0.0001 and a batch size of 64.

5 Results

GLUE: Table 1 shows results on downstream GLUE tasks and model sizes for our proposed models, BERT$_{\text{BASE/LARGE}}$, and baselines. Our models consistently improve upon the identically parameterized *NoKD* baselines, indicating mixed-vocabulary training is better than training from scratch and avoids a large teacher-student performance gap. Compared with PKD/DistilBERT, our 6-layer model outperforms PKD$_3$ while being >7x smaller and our 12-layer model is comparable to PKD$_6$ and DistilBERT$_4$ while being ∼5-6x smaller.

Interestingly, our models do particularly well on the MRPC task: the 6-layer distilled model performs almost as well as PKD$_6$ while being over 10x smaller. This may be due to our smaller models being data-efficient on the smaller MRPC dataset.

TinyBERT and Bert-of-Theseus are trained in task-specific fashion i.e., a teacher model already finetuned on the downstream task is used for distillation. TinyBERT's non-task-specific model results are reported on GLUE dev sets: these results are, therefore, not directly comparable with ours. Even so, our 12-layer model performs credibly

Model	#Params	Latency	Intent Acc	Slot F1
$\text{BERT}_{\text{BASE}}$ (Chen et al., 2019a)	109M	340ms	98.6	97.0
StackProp (Qin et al., 2019)	2.6M	>70ms	98.0	94.2
Mixed-vocab distilled, L-6, H-96	1.2M	6ms	98.9	92.8
Mixed-vocab distilled, L-6, H-192	3.6M	14ms	98.8	94.6
Mixed-vocab distilled, L-6, H-256	6.2M	20ms	98.7	95.0

Table 2: Results on the SNIPS dataset. Latency is measured with 4 CPU threads on a Pixel 4 mobile device.

compared with the two, presenting a competitive size-accuracy tradeoff, particularly when compared to the 6x larger BERT-of-Theseus.

MobileBERT performs strongly for the size while being task-agnostic. Our 12-layer model, in comparison, retains ∼98% of its performance with 57% fewer parameters and may thus be better-suited for use on highly resource-limited devices.

TinyBERT sees major gains from task-specific data augmentation and distillation, and MobileBERT from student architecture search and bottleneck layers. Notably, our technique targets the student vocabulary without conflicting with any of the above methods and can, in fact, be combined with these methods for even smaller models.

SNIPS: Table 2 shows results on the SNIPS intent and slot tasks for our models and two state-of-the-art baselines. Our smallest 6-layer model retains over 95% of the $\text{BERT}_{\text{BASE}}$ model's slot filling F1 score (Sang and Buchholz, 2000) while being 30x smaller ($<$ 10 MB w/o quantization) and 57x faster on a mobile device, yet task-agnostic. Our other larger distilled models also demonstrate strong performance (0.2-0.5% slot F1 higher than the respective *NoKD* baselines) with small model sizes and latencies low enough for real-time inference. This indicates that small multi-task BERT models (Tsai et al., 2019) present better trade-offs for on-device usage for size, accuracy and latency versus recurrent encoder-based models such as StackProp.

6 Discussion

Impact of vocabulary size: We trained a model from scratch identical to $\text{BERT}_{\text{BASE}}$ except with our 5K-WP student vocabulary. On the SST-2 and MNLI-m dev sets, this model obtained 90.9% and 83.7% accuracy respectively - only 1.8% and 0.7% lower respectively compared to $\text{BERT}_{\text{BASE}}$.

Since embeddings account for a larger fraction of model parameters with fewer layers, we trained another model identical to our 6×256 model, but with a 30K-WP vocabulary and teacher label distillation. This model showed small gains (0.1% / 0.5% accuracy on SST-2 / MNLI-m dev) over our analogous distilled model, but with 30% more parameters solely due to the larger vocabulary.

This suggests that a small WordPiece vocabulary may be almost as effective for sequence classification/tagging tasks, especially for smaller BERT models and up to moderately long inputs. Curiously, increasing the student vocabulary size to 7K or 10K did not lead to an increase in performance on GLUE. We surmise that this may be due to underfitting owing to the embeddings accounting for a larger proportion of the model parameters.

Alternative vocabulary pruning: Probing other strategies for a small-vocabulary model, we used the above 6×256 30K-WP vanilla distilled model to obtain a smaller model by pruning the vocabulary to contain the intersection of the 30K and 5K vocabularies (total 4629 WPs). This model is 1.2% smaller than our 4928-WP distilled model, but drops 0.8% / 0.7% on SST-2/MNLI-m dev sets.

Furthermore, to exploit the high overlap in vocabularies, we tried running our distillation pipeline but with the embeddings for student tokens (after projecting up to the teacher dimension) also present in the teacher vocabulary tied to the teacher embeddings for those tokens. This model, however, dropped 0.7% / 0.5% on SST-2/MNLI-m compared to our analogous 6×256 distilled model.

We also tried pretraining $\text{BERT}_{\text{LARGE}}$ from scratch with the 5K vocabulary and doing vanilla distillation for a 6×256 student: this model dropped 1.2% / 0.7% for SST-2/MNLI-m over our similar distilled model, indicating the efficacy of mixed-vocabulary training over vanilla distillation.

7 Conclusion

We propose a novel approach to knowledge distillation for BERT, focusing on using a significantly smaller vocabulary for the student BERT models. Our *mixed-vocabulary training* method encourages implicit alignment of the teacher and student WordPiece embeddings. Our highly-compressed 6 and 12-layer distilled student models are optimized for on-device use cases and demonstrate competitive performance on both benchmark datasets and practical tasks. Our technique is unique in targeting the student vocabulary size, enabling easy combination with most BERT distillation methods.

References

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662.

Daoyuan Chen, Yaliang Li, Minghui Qiu, Zhen Wang, Bofang Li, Bolin Ding, Hongbo Deng, Jun Huang, Wei Lin, and Jingren Zhou. 2020. Adabert: Task-adaptive bert compression with differentiable neural architecture search. *arXiv preprint arXiv:2001.04246*.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019a. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Wenhu Chen, Yu Su, Yilin Shen, Zhiyu Chen, Xifeng Yan, and William Yang Wang. 2019b. How large a vocabulary does text classification need? a variational approach to vocabulary selection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3487–3497.

Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. 2015. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pages 2285–2294.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

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*.

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

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

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

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

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*.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2019. Bpe-dropout: Simple and effective subword regularization. *arXiv preprint arXiv:1910.13267*.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2078–2087.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

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

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*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2019. Q-bert: Hessian based ultra low precision quantization of bert. *arXiv preprint arXiv:1909.05840*.

Raphael Shu and Hideki Nakayama. 2018. Compressing word embeddings via deep compositional code learning. In *International Conference on Learning Representations*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical bert models for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3623–3627.

Andrew Tulloch and Yangqing Jia. 2017. High performance ultra-low-precision convolutions on mobile devices. *arXiv preprint arXiv:1712.02427*.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

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

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

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. Deebert: Dynamic early exiting for accelerating bert inference. *arXiv preprint arXiv:2004.12993*.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. Bert-of-theseus: Compressing bert by progressive module replacing. *arXiv preprint arXiv:2002.02925*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, and Cho-Jui Hsieh. 2019. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*.

Sergey Zagoruyko and Nikos Komodakis. 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*.

Yiren Zhou, Seyed-Mohsen Moosavi-Dezfooli, Ngai-Man Cheung, and Pascal Frossard. 2018. Adaptive quantization for deep neural network. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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 *Proceedings of the IEEE international conference on computer vision*, pages 19–27.