

# EXTREME LANGUAGE MODEL COMPRESSION WITH OPTIMAL SUBWORDS AND SHARED PROJECTIONS

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Pre-trained deep neural network language models such as ELMo, GPT, BERT and XLNet have recently achieved state-of-the-art performance on a variety of language understanding tasks. However, their size makes them impractical for a number of scenarios, especially on mobile and edge devices. In particular, the input word embedding matrix accounts for a significant proportion of the model’s memory footprint, due to the large input vocabulary and embedding dimensions. Knowledge distillation techniques have had success at compressing large neural network models, but they are ineffective at yielding student models with vocabularies different from the original teacher models. We introduce a novel knowledge distillation technique for training a student model with a significantly smaller vocabulary as well as lower embedding and hidden state dimensions. Specifically, we employ a dual-training mechanism that trains the teacher and student models simultaneously to obtain optimal word embeddings for the student vocabulary. We combine this approach with learning shared projection matrices that transfer layer-wise knowledge from the teacher model to the student model. Our method is able to compress the BERT<sub>BASE</sub> model by more than 60x, with only a minor drop in downstream task metrics, resulting in a language model with a footprint of under 7MB. Experimental results also demonstrate higher compression efficiency and accuracy when compared with other state-of-the-art compression techniques.

## 1 INTRODUCTION

Recently, contextual-aware language models such as ELMo (Peters et al., 2018), GPT (Radford et al., 2019), BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019) have shown to greatly outperform traditional word embedding models including Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) in a variety of NLP tasks. These pre-trained language models, when fine-tuned on downstream language understanding tasks such as sentiment classification (Socher et al., 2013), natural language inference (Williams et al., 2018) and reading comprehension (Rajpurkar et al., 2016; Lai et al., 2017), have achieved state-of-the-art performance. However, the large number of parameters in these models, often above hundreds of millions, makes it impossible to host them on resource-constrained tasks such as doing real-time inference on mobile and edge devices.

Besides utilizing model quantization techniques (Gong et al., 2014; Lin et al., 2016) which aim to reduce the floating-point accuracy of the parameters, significant recent research has focused on knowledge distillation (Ba & Caruana, 2014; Hinton et al., 2015) techniques. Here, the goal is to train a small-footprint student model by borrowing knowledge, such as through a soft predicted label distribution, from a larger pre-trained teacher model.

However, a significant bottleneck that has been overlooked by previous efforts is the input vocabulary size and its corresponding word embedding matrix, often accounting for a significant proportion of all model parameters. For instance, the embedding table of the BERT<sub>BASE</sub> model, comprising over 30K WordPiece tokens (Wu et al., 2016b), accounts for over 21% of the model size. While there has been existing work on reducing NLP model vocabulary sizes (Sennrich et al., 2016), distillation techniques cannot utilize these, since they require the student and teacher models to share the same vocabulary and output space. This profoundly limits their potential to further reduce model sizes.

We present two novel ideas to improve the effectiveness of knowledge distillation, in particular for BERT, with the focus on bringing down model sizes to as much as a few mega-bytes. Our model

is among the first to propose to use a significantly smaller vocabulary for the student model learned during distillation. In addition, instead of distilling solely on the teacher model’s final-layer outputs, our model leverages layer-wise teacher model parameters to directly optimize the parameters of the corresponding layers in the student model. Specifically, our contributions are:

- **Dual Training:** Our teacher and student models have different vocabularies and incompatible tokenizations for the same sequence. To address this during distillation, we feed the teacher model a mix of teacher vocabulary-tokenized and student vocabulary-tokenized words within a single sequence. Coupled with the masked language modeling task, this encourages an implicit alignment of the teacher and student WordPiece embeddings, since the student vocabulary embedding may be used as context to predict a word tokenized by the teacher vocabulary and vice versa.
- **Shared Variable Projections:** To minimize the loss of information from reducing the hidden state dimension, we introduce a separate loss to align the teacher and student models’ trainable variables. This allows for more direct layer-wise transfer of knowledge to the student model.

Using the combination of dual training and shared variable projections, we train a 12-layer highly-compressed student BERT model, achieving a maximum compression ratio of  $\sim 61.94x$  (with 48 dimension size) compared to the teacher BERT<sub>BASE</sub> model. We conduct experiments for measuring both generalized language modeling perspective and for downstream tasks, demonstrating competitive performance with high compression ratios for both families of tasks.

## 2 RELATED WORK

Research in neural network model compression has been concomitant with the rise in popularity of neural networks themselves, since these models have often been memory-intensive for the hardware of their time. Work in model compression for NLP applications falls broadly into four categories: matrix approximation, parameter pruning/sharing, weight quantization and knowledge distillation.

A family of approaches (Sindhwani et al., 2015; Tulloch & Jia, 2017) seeks to compress the matrix parameters of the models by low-rank approximation i.e. the full-rank matrix parameter is approximated using multiple low-rank matrices, thereby reducing the effective number of model parameters. Another line of work explores parameter pruning and sharing-based methods (Li et al., 2016; Luo et al., 2017; Anwar et al., 2017; See et al., 2016), which explore the redundancy in model parameters and try to remove redundant weights as well as neurons, for a variety of neural network architectures.

Model weight quantization techniques (Chen et al., 2015; Lin et al., 2017; Wu et al., 2016a; Zhou et al., 2018) focus on mapping model weights to lower-precision integers and floating-point numbers. These can be especially effective with hardware supporting efficient low-precision calculations. More recently, Shen et al. (2019) apply quantization to BERT-based transformer models.

Knowledge distillation (Ba & Caruana, 2014; Hinton et al., 2015) differs from the other discussed approaches: the smaller student model may be parametrized differently from the bigger teacher model, affording more modeling freedom. Teaching a student model to match the soft output label distributions from a larger model alongside the hard ground-truth distribution works well for many tasks, such as machine translation (Kim & Rush, 2016) and language modeling (Yu et al., 2018).

Not limited to the teacher model outputs, some approaches perform knowledge distillation via attention transfer (Zagoruyko & Komodakis, 2016), or via feature maps or intermediate model outputs (Romero et al., 2014; Yim et al., 2017; Huang & Wang, 2017). More relevant to current work, Tang et al. (2019) and Sun et al. (2019) employ variants of these techniques to BERT model compression by reducing the number of transformer layers. However, as explained before, these approaches are not immediately applicable to our setting due to incompatible teacher and student model vocabularies, and do not focus sufficiently on the embedding matrix size (Joulin et al., 2016).

## 3 METHODOLOGY

Our knowledge distillation approach is centered around reducing the number of WordPiece tokens in the model vocabulary. In this section, we first discuss the rationale behind this reduction and the challenges it introduces, followed by our techniques, namely *dual training* and *shared projection*.

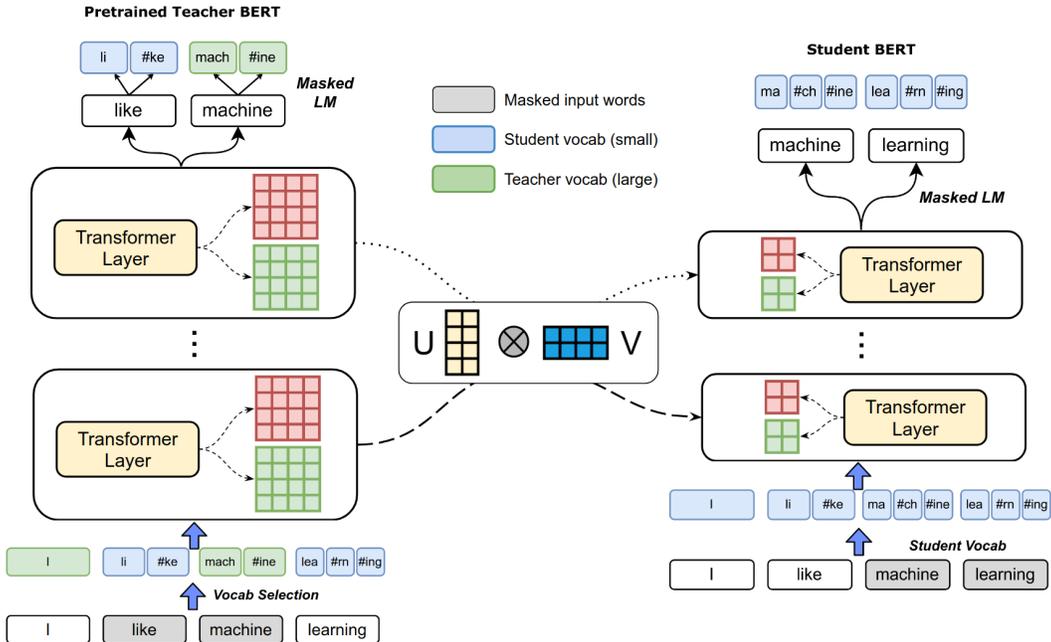


Figure 1: Knowledge Distillation on BERT with smaller student vocabulary. (Left) A pre-trained teacher BERT model with default BERT parameters (e.g., 30K vocab, 768 hidden state dimension). (Right) A student BERT model trained from scratch with smaller vocab (5K) and hidden state dimension (e.g., 48). During distillation, the teacher model randomly selects a vocabulary to segment each input word. The red and green square next to the transformer layers indicate trainable parameters for both the student and teacher models - note that our student models have smaller model dimensions. The projection matrices  $U$  and  $V$ , shown as having representative shapes, are shared across all layers for model parameters that have the same dimensions.

### 3.1 OPTIMAL SUBWORD EMBEDDINGS VIA KNOWLEDGE DISTILLATION

We follow the general knowledge distillation paradigm of training a small student model from a large teacher model. Our teacher model is a 12-layer uncased BERT<sub>BASE</sub>, trained with 30522 WordPiece tokens and 768-dimensional embeddings and hidden states. We denote the teacher model parameters by  $\theta_t$ . Our student model consists of an equal number of transformer layers with parameters denoted by  $\theta_s$ , but with a smaller vocabulary as well as embedding/hidden dimensions, illustrated in Figure 1. Using the same WordPiece algorithm and training corpus as BERT, we obtain a vocabulary of 4928 WordPieces, which we use for the student model.

WordPiece tokens (Wu et al., 2016b) are sub-word units obtained by applying a greedy segmentation algorithm to the training corpus: a desired number (say,  $D$ ) of WordPieces are chosen such that the segmented corpus is minimal in the number of WordPieces used. A cursory look at both vocabularies reveals that 93.9% of the WordPieces in the student vocabulary also exist in the teacher vocabulary, suggesting room for a reduction in the WordPiece vocabulary size from 30K tokens.

Since we seek to train a general-purpose student language model, we elect to reuse the teacher model’s original training objective to optimize the student model, i.e., masked language modeling and next sentence prediction, before any fine-tuning. In the former task, words in context are randomly masked, and the language model needs to predict those words given the masked context. In the latter task, given a pair of sentences, the language model predicts whether the pair is consistent.

However, since the student vocabulary is not a complete subset of the teacher vocabulary, the two vocabularies may tokenize the same words differently. As a result, the outputs of the teacher and student model for the masked language modeling task may not align. Even with the high overlap between the two vocabularies, the need to train the student embedding from scratch, and the change

in embedding dimension precludes existing knowledge distillation techniques, which rely on the alignment of both models’ output spaces. As a result, we explore two alternative approaches that enable implicit transfer of knowledge to the student model, which we describe below.

### 3.2 DUAL TRAINING

During distillation, for a given training sequence input to the teacher model, we propose to mix the teacher and student vocabularies by randomly selecting (with a probability  $p_{DT}$ , a hyperparameter) tokens from the sequence to segment using the student vocabulary, with the other tokens segmented using the teacher vocabulary. As illustrated in Figure 1, given the input context [ ‘I’, ‘like’, ‘machine’, ‘learning’ ], the words ‘I’ and ‘machine’ are segmented using the teacher vocabulary (in green), while ‘like’ and ‘learning’ are segmented using the student vocabulary (in blue). Similar to cross-lingual training in Lample & Conneau (2019), this encourages alignment of the representations for the same word as per the teacher and student vocabularies. This is effected through the masked language modeling task: the model now needs to learn to predict words from the student vocabulary using context words segmented using the teacher vocabulary, and vice versa.

The expectation is that the student embeddings can be learned effectively this way from the teacher embeddings as well as model parameters  $\theta_t$ . Note that we perform dual training only for the teacher model inputs: the student model receives words segmented exclusively using the student vocabulary. Also, during masked language modeling, the model uses different softmax layers for the teacher and the student vocabularies depending on which one was used to segment the word in question.

### 3.3 SHARED PROJECTIONS

Relying solely on teacher model outputs to train the student model may not generalize well (Sun et al., 2019). Therefore, some approaches utilize and try to align the student model’s intermediate predictions to those of the teacher (Romero et al., 2014). In our setting, however, since the student and teacher model output spaces are not identical, intermediate model outputs may prove hard to align. Instead, we seek to directly minimize the loss of information from the teacher model parameters  $\theta_t$  to the student parameters  $\theta_s$  with smaller dimensions. We achieve this by projecting the model parameters into the same space, to encourage alignment.

More specifically, as in Figure 1, we project each trainable variable in  $\theta_t$  to the same shape as the corresponding variable in  $\theta_s$ . For example, for all the trainable variables  $\theta_t$  with shape  $768 \times 768$ , we learn two projection matrices  $\mathbf{U} \in \mathbb{R}^{d \times 768}$  and  $\mathbf{V} \in \mathbb{R}^{768 \times d}$  to project them into the corresponding space of the student model variable  $\theta'_t$ , where  $d$  is the student model’s hidden dimension.  $\mathbf{U}$  and  $\mathbf{V}$  are common to all BERT model parameters of that dimensionality; in addition,  $\mathbf{U}$  and  $\mathbf{V}$  are not needed for fine-tuning or inference after distillation. In order to align the student variable and the teacher variable’s projection, we introduce a separate mean square error loss defined in Equation 1, where  $\downarrow$  stands for *down projection* (since the projection is to a lower dimension).

$$L_p^\downarrow = \sum_{\theta'_t, \theta'_s \subset \theta_t, \theta_s} \|\mathbf{U}\theta'_t\mathbf{V} - \theta'_s\|^2 \quad (1)$$

The above loss function aligns the trainable variables in the student space. Alternatively, we can project trainable variables in  $\theta_s$  to the same shape as in  $\theta_t$ . This way, the loss function in Equation 2, ( $\uparrow$  denotes *up projection*) can compare the trainable variables in the teacher space.

$$L_p^\uparrow = \sum_{\theta'_t, \theta'_s \subset \theta_t, \theta_s} \|\theta'_t - \mathbf{V}\theta'_s\mathbf{U}\|^2 \quad (2)$$

### 3.4 OPTIMIZATION OBJECTIVE

Our final loss function includes, in addition to an optional projection loss, masked language modeling cross-entropy losses for the student as well as the teacher models, since the teacher model is trained with dual-vocabulary inputs and is not static.  $P(y_i = c|\theta_s)$  and  $P(y_i = c|\theta_t)$  denote the stu-

dent and teacher model prediction probabilities for class  $c$  respectively, and  $\mathbb{1}$  denotes an indicator function. Equations 3 and 4 below define the final loss  $L_{final}$ , where  $\epsilon$  is a hyperparameter.

$$L_{ce} = \sum_i \left( \sum_{c \in C} [\mathbb{1}_{[y_i]=c} \log P(y_i = c | \theta_s)] + \sum_{c \in C} [\mathbb{1}_{[y_i]=c} \log P(y_i = c | \theta_t)] \right) \quad (3)$$

$$L_{final} = L_p + \epsilon \times L_{ce} \quad (4)$$

## 4 EXPERIMENTS

To evaluate our knowledge distillation approach, we design two classes of experiments. First, we evaluate the distilled student language models using the masked word prediction task on an unseen evaluation corpus, for an explicit evaluation of the language model. Second, we fine-tune the language model by adding a task-specific affine layer on top of the student language model outputs, on a suite of downstream sentence and sentence pair classification tasks. This is meant to be an implicit evaluation of the quality of the representations learned by the student language model. We describe these experiments, along with details on training, implementation and our baselines below.

### 4.1 LANGUAGE MODEL TRAINING

During the distillation of the teacher BERT model to train the student BERT language model, we utilize the same corpus as was used to train the teacher i.e. BooksCorpus (Zhu et al., 2015) and English Wikipedia, with whitespaces used to tokenize the text into words. We only use the masked language modeling task to calculate the overall distillation loss from Section 3.3, since the next sentence prediction loss hurt performance slightly. Dual training is enabled for teacher model inputs, with  $p_{DT}$ , the probability of segmenting a teacher model input word using the student vocabulary, set to 0.5. For experiments including shared projection, the projection matrices  $\mathbf{U}$  and  $\mathbf{V}$  utilized Xavier initialization (Glorot & Bengio, 2010). The loss weight coefficient  $\epsilon$  is set to 1 after tuning.

It is worth noting that in contrast to a number of existing approaches, we directly distill the teacher BERT language model, not yet fine-tuned on a downstream task, to obtain a student language model that is task-agnostic. For downstream tasks, we fine-tune this distilled student language model.

Distillation is carried out on Cloud TPUs in a 4x4 pod configuration<sup>1</sup> (32 TPU cores overall). We optimized the loss using LAMB (You et al., 2019) for 250K steps, with a learning rate of 0.00125 and batch size of 4096. Depending on the student model dimension, training took between 2-4 days.

### 4.2 MODELS AND BASELINES

We evaluate three variants of our distilled student models: with only dual training of the teacher and student vocabularies (*DualTrain*) and with dual training along with down-projection (*DualTrain + SharedProjDown*) or up-projection (*DualTrain + SharedProjUp*) of the teacher model parameters. For each of these configurations, we train student models with embedding and hidden dimensions 48, 96 and 192, for 9 total variants, each using a compact 5K-WordPiece vocabulary. Table 1 presents some statistics on these models’ sizes: our smallest model contains two orders of magnitude fewer parameters, and requires only 1% floating-point operations when compared to the BERT<sub>BASE</sub> model.

For the language modeling evaluation, we also evaluate a baseline without knowledge distillation (termed *NoKD*), with a model parameterized identically to the distilled student models but trained directly on the teacher model objective from scratch. For downstream tasks, we compare with *NoKD* as well as Patient Knowledge Distillation (*PKD*) from Sun et al. (2019), who distill the 12-layer BERT<sub>BASE</sub> model into 3 and 6-layer BERT models by using the teacher model’s hidden states.

<sup>1</sup><https://cloudplatform.googleblog.com/2018/06/Cloud-TPU-now-offers-preemptible-pricing-and-global-availability.html>

Model Type	Vocab Size	Hidden Dim	# Params	Model Size (MB)	FLOPS ratio
BERT <sub>DISTILLED</sub>	4928	48	1,775,910	6.8	1.3%
		96	5,665,926	22	1.32%
		192	19,169,094	73	4.49%
BERT <sub>BASE</sub>	30522	768	110,106,428	420	100%

Table 1: A summary of our student models’ sizes compared to BERT<sub>BASE</sub>. #Params indicates the number of parameters in the student model, model size is measured in megabytes, and FLOPS ratio measures the relative ratio of floating point operations required for inference on the model.

### 4.3 EVALUATION TASKS AND DATASETS

For explicit evaluation of the generalized language perspective of the distilled student language models, we use the Reddit dataset (Al-Rfou et al., 2016)<sup>2</sup> to measure word mask prediction accuracy of the student models, since the language used on Reddit is different from that in the training corpora. The dataset is preprocessed similarly to the training corpora, except we do not need to tokenize it using the teacher vocabulary, since we only run and evaluate the student models.

For implicit evaluation on downstream language understanding tasks, we fine-tune and evaluate the distilled student models on three tasks from the GLUE benchmark (Wang et al., 2019):

- Stanford Sentiment Treebank (SST-2) (Socher et al., 2013), a two-way sentence sentiment classification task with 67K training instances,
- Microsoft Research Paraphrase Corpus (MRPC) (Dolan & Brockett, 2005), a two-way sentence pair classification task to identify paraphrases, with 3.7K training instances, and
- Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2018), a three-way sentence pair classification task with 393K training instances, to identify premise-hypothesis relations. There are separate development and test sets for genre-matched and genre-mismatched premise-hypothesis pairs; we tune our models solely on the genre-matched development set.

For all downstream task evaluations, we fine-tune for 10 epochs using LAMB with a learning rate of 0.0002 and batch size of 32. Since our language models are trained with a maximum sequence length of 128 tokens, we do not evaluate on reading comprehension datasets such as SQuAD (Rajpurkar et al., 2016) or RACE (Lai et al., 2017), which require models supporting longer sequences.

Model ↓ Hidden dimension →	48	96	192
NoKD Baseline	28.83	33.70	40.01
DualTrain	29.51	34.59	40.58
DualTrain + SharedProjDown	29.55	34.71	40.59
DualTrain + SharedProjUp	<b>29.89</b>	<b>34.74</b>	<b>40.64</b>

Table 2: Masked language modeling task accuracy for the distilled student models and a fine-tune-from-scratch baseline. We observe consistently better performance for our proposed approaches.

## 5 RESULTS

Table 2 contains masked word prediction accuracy figures for the different models and the *NoKD* baseline. We observe that dual training significantly improves over the baseline for all model dimensions, and that both shared projection losses added to dual training further improve the word prediction accuracy. It is interesting to note that for all model dimensions, *SharedProjUp* projecting into the teacher space outperforms *SharedProjDown*, significantly so for dimension 48. Expectedly, there is a noticeable performance drop going from 192 to 96 to 48-dimensional hidden state models.

<sup>2</sup>We use the test partition of the dataset, which is 6,400 instances.

Model	Hidden Dim	Vocab Size	Compress Factor	MRPC (F1/Acc)	MNLI-m (Acc)	MNLI-mm (Acc)	SST-2 (Acc)
Teacher BERT <sub>BASE</sub>	768	30522	1x	88.5/84.3	84.0	82.8	93.5
PKD, 6 layers (Sun et al., 2019)	768	30522	1.64x	85.0/79.9	81.5	81.0	92.0
PKD, 3 layers (Sun et al., 2019)			2.40x	80.7/72.5	76.7	76.3	87.5
NoKD Baseline	192	4928	5.74x	82.6/74.1	77.4	76.5	87.1
DualTrain				82.5/76.6	78.1	77.3	<b>88.4</b>
DualTrain + SharedProjDown				83.6/76.9	<b>78.2</b>	<b>77.7</b>	<b>88.4</b>
DualTrain + SharedProjUp				<b>84.9/78.5</b>	77.5	76.7	88.0
NoKD Baseline	96	4928	19.41x	84.6/77.3	76.2	75.1	85.4
DualTrain				<b>86.1/80.5</b>	76.1	74.7	85.4
DualTrain + SharedProjDown				83.7/77.5	<b>76.5</b>	<b>75.2</b>	<b>85.6</b>
DualTrain + SharedProjUp				84.9/78.1	76.4	<b>75.2</b>	84.7
NoKD Baseline	48	4928	61.94x	76.3/66.1	70.9	70.2	79.5
DualTrain				77.5/66.8	70.6	69.9	79.8
DualTrain + SharedProjDown				78.0/68.2	<b>71.3</b>	70.4	80.0
DualTrain + SharedProjUp				<b>79.3/68.6</b>	71.0	<b>70.8</b>	<b>82.2</b>

Table 3: Results of the distilled models, the teacher model and baselines on the downstream language understanding task test sets, obtained from the GLUE server, along with the size parameters and compression ratios of the respective models compared to the teacher BERT<sub>BASE</sub>. MNLI-m and MNLI-mm refer to the genre-matched and genre-mismatched test sets for MNLI.

Note that because of the differing teacher and student model vocabularies, masked word prediction accuracy for the teacher BERT<sub>BASE</sub> model is not directly comparable with the student models.

Table 3 shows results on the downstream language understanding tasks, as well as model sizes, for our approaches, the BERT<sub>BASE</sub> teacher model, and the *PKD* and *NoKD* baselines. We note that models trained with our proposed approaches perform strongly and consistently improve upon the identically parametrized *NoKD* baselines, indicating that the dual training and shared projection techniques are effective, without incurring significant losses against the BERT<sub>BASE</sub> teacher model. Comparing with the *PKD* baseline, our 192-dimensional models, achieving a higher compression rate than either of the *PKD* models, perform better than the 3-layer *PKD* baseline and are competitive with the larger 6-layer baseline on task accuracy while being nearly 5 times as small.

Another observation we make is that the performance drop from 192-dimensional to 96-dimensional models is minimal (less than 2% for most tasks). For the MRPC task, in fact, the 96-dimensional model trained with dual training achieves an accuracy of 80.5%, which is higher than even the *PKD* 6-layer baseline with nearly 12 times as many parameters. Finally, our highly-compressed 48-dimensional models also perform respectably: the best 48-dimensional models are in a similar performance bracket as the 3-layer *PKD* model, a model 25 times larger by memory footprint.

## 6 DISCUSSION

**Shared projections and model performance:** We see that for downstream task performance, dual training still consistently improves upon the direct fine-tuning approach for virtually all experiments. The effect of shared variable projection, however, is less pronounced, with consistent improvements visible only for MRPC and for the 48-dimensional models i.e. the smallest dataset and models respectively in our experiments. This aligns with our intuition for variable projection as a more direct way to provide a training signal from the teacher model internals, which can assume more importance for a low-data or small-model scenario. However, for larger models and more data, the linear projection of parameters may be reducing the degrees of freedom available to the model, since linear projection is a fairly simple function to align the teacher and student parameter spaces.

A related comparison of interest is between up-projection and down-projection of the model variables: we note up-projection does visibly better on the language modeling task and slightly better

on the downstream tasks. The parameters of a well-trained teacher model represent a high-quality local minimum in the teacher space, which may be easier to search for during up-projection.

**Vocabulary size tradeoffs:** Issues with input vocabulary size are peculiar to problems in natural language processing: they do not always apply to other areas such as computer vision, where a small fixed number of symbols can encode most inputs. There has been some work on reducing input vocabulary sizes for NLP, but typically not targeting model compression. One concern with reducing the vocabularies of NLP models is it pushes the average tokenized sequence lengths up, making model training harder. In this work, however, we consider classification tasks on shorter texts, which are not as affected by input sequence lengths as, say, tasks such as machine translation are. Furthermore, many real-world applications revolve around short text inputs, which is why a better trade-off between vocabulary size and sequence lengths may be worthwhile for such applications.

**Order of distillation and fine-tuning:** Most of the existing work on distilling language models such as BERT and reporting results on downstream tasks, including some of the baselines in this work, first fine-tune a teacher model on the downstream tasks, and then distill this model. Our goal in this work, however, is to explore the limits to which BERT’s language modeling capacity itself, and how much of it is driven by its large WordPiece vocabulary. We leave experiments on distilling fine-tuned teacher models, potentially yielding better results on downstream tasks, to future work.

## 7 CONCLUSION

We proposed two novel ideas to improve the effectiveness of knowledge distillation for BERT, focusing on using a significantly smaller vocabulary, as well as smaller embedding and hidden dimensions for the student BERT language models. Our *dual training* mechanism encourages implicit alignment of the teacher and student WordPiece embeddings, and *shared variable projection* allows for the faster and direct layer-wise transfer of knowledge to the student BERT model. Combining the two techniques, we trained a series of highly-compressed 12-layer student BERT models. Experiments on these models, to evaluate both generalized language perspective and four standardized downstream tasks, demonstrate the effectiveness of our proposed methods on both model accuracy and compression efficiency.

One future direction of interest is to combine our approach with existing work to reduce the number of layers in the student models and explore other approaches such as low-rank matrix factorization to transfer model parameters from the teacher space to the student space. In addition, taking into account the frequency distribution of the WordPiece tokens while training embeddings may help optimize the model size further.

## REFERENCES

- Rami Al-Rfou, Marc Pickett, Javier Snaider, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. Conversational contextual cues: The case of personalization and history for response ranking. *arXiv preprint arXiv:1606.00372*, 2016.
- Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):32, 2017.
- Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pp. 2654–2662, 2014.
- Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pp. 2285–2294, 2015.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing*, 2005.

- Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.
- Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*, 2014.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, H erve J egou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.
- Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*, 2016.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, 2017.
- Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.
- Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.
- Darryl Lin, Sachin Talathi, and Sreekanth Annapureddy. Fixed point quantization of deep convolutional networks. In *International Conference on Machine Learning*, pp. 2849–2858, 2016.
- Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. In *Advances in Neural Information Processing Systems*, pp. 345–353, 2017.
- Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pp. 5058–5066, 2017.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 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)*, pp. 2227–2237, 2018.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016.
- Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

- Abigail See, Minh-Thang Luong, and Christopher D Manning. Compression of neural machine translation models via pruning. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 291–301, 2016.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 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)*, pp. 1715–1725, 2016.
- Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-bert: Hessian based ultra low precision quantization of bert. *arXiv preprint arXiv:1909.05840*, 2019.
- Vikas Sindhvani, Tara Sainath, and Sanjiv Kumar. Structured transforms for small-footprint deep learning. In *Advances in Neural Information Processing Systems*, pp. 3088–3096, 2015.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.
- Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019.
- Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*, 2019.
- Andrew Tulloch and Yangqing Jia. High performance ultra-low-precision convolutions on mobile devices. *arXiv preprint arXiv:1712.02427*, 2017.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=rJ4km2R5t7>.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 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)*, pp. 1112–1122, 2018.
- Xundong Wu, Yong Wu, and Yong Zhao. Binarized neural networks on the imagenet classification task. *arXiv preprint arXiv:1604.03058*, 2016a.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016b.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.
- Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4133–4141, 2017.
- Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.
- Seunghak Yu, Nilesh Kulkarni, Haejun Lee, and Jihie Kim. On-device neural language model based word prediction. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pp. 128–131, 2018.

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

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

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 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*, pp. 19–27, 2015.