Distilling Task-Specific Knowledge from BERT into Simple Neural Networks

Raphael Tang[*], **Yao Lu**[*], **Linqing Liu**[*], **Lili Mou**, **Olga Vechtomova**, and **Jimmy Lin**
University of Waterloo
{r33tang, yao.lu, linqing.liu}@uwaterloo.ca
doublepower.mou@gmail.com {ovechtom, jimmylin}@uwaterloo.ca

Abstract

In the natural language processing literature, neural networks are becoming increasingly deeper and complex. The recent poster child of this trend is the deep language representation model, which includes BERT, ELMo, and GPT. These developments have led to the conviction that previous-generation, shallower neural networks for language understanding are obsolete. In this paper, however, we demonstrate that rudimentary, lightweight neural networks can still be made competitive *without* architecture changes, external training data, or additional input features. We propose to distill knowledge from BERT, a state-of-the-art language representation model, into a single-layer BiLSTM, as well as its siamese counterpart for sentence-pair tasks. Across multiple datasets in paraphrasing, natural language inference, and sentiment classification, we achieve comparable results with ELMo, while using roughly 100 times fewer parameters and 15 times less inference time.

1 Introduction

In the natural language processing (NLP) literature, the march of the neural networks has been an unending yet predictable one, with new architectures constantly surpassing previous ones in not only performance and supposed insight but also complexity and depth. In the midst of all this neural progress, it becomes easy to dismiss earlier, "first-generation" neural networks as obsolete. Ostensibly, this appears to be true: Peters et al. (2018) show that using pretrained deep word representations achieves state of the art on a variety of tasks. Recently, Devlin et al. (2018) have pushed this line of work even further with bidirectional encoder representations from transformers (BERT), deeper models that greatly improve state of the art on more tasks. More recently, OpenAI has described GPT-2, a state-of-the-art, larger transformer model trained on even more data.[1]

Such large neural networks are, however, problematic in practice. Due to the large number of parameters, BERT and GPT-2, for example, are undeployable in resource-restricted systems such as mobile devices. They may be inapplicable in real-time systems either, because of low inference-time efficiency. Furthermore, the continued slowdown of Moore's Law and Dennard scaling (Han, 2017) suggests that there exists a point in time when we must compress our models and carefully evaluate our choice of the neural architecture.

In this paper, we propose a simple yet effective approach that transfers task-specific knowledge from BERT to a shallow neural architecture—in particular, a bidirectional long short-term memory network (BiLSTM). Our motivation is twofold: we question whether a simple architecture actually lacks representation power for text modeling, and we wish to study effective approaches to transfer knowledge from BERT to a BiLSTM. Concretely, we leverage the knowledge distillation approach (Ba and Caruana, 2014; Hinton et al., 2015), where a larger model serves as a *teacher* and a small model learns to mimic the teacher as a *student*. This approach is model agnostic, making knowledge transfer possible between BERT and a different neural architecture, such as a single-layer BiLSTM, in our case.

To facilitate effective knowledge transfer, however, we often require a large, unlabeled dataset. The teacher model provides the probability logits and estimated labels for these unannotated samples, and the student network learns from the teacher's outputs. In computer vision, unlabeled images are usually easy to obtain through augmenting the data using rotation, additive noise,

[*]Equal contribution. Ordering decided by coin toss.

[1] https://goo.gl/Frmwqe

and other distortions. However, obtaining additional, even unlabeled samples for a specific task can be difficult in NLP. Traditional data augmentation in NLP is typically task-specific (Wang and Eisner, 2016; Serban et al., 2016) and difficult to extend to other NLP tasks. To this end, we further propose a novel, rule-based textual data augmentation approach for constructing the knowledge transfer set. Although our augmented samples are not fluent natural language sentences, experimental results show that our approach works surprisingly well for knowledge distillation.

We evaluate our approach on three tasks in sentence classification and sentence matching. Experiments show that our knowledge distillation procedure significantly outperforms training the original simpler network alone. To our knowledge, we are the first to explore distilling knowledge from BERT. With our approach, a shallow BiLSTM-based model achieves results comparable to Embeddings from Language Models (ELMo; Peters et al., 2018), but uses around 100 times fewer parameters and performs inference 15 times faster. Therefore, our model becomes a state-of-the-art "small" model for neural NLP.

2 Related Work

In the past, researchers have developed and applied various neural architectures for NLP, including convolutional neural networks (Kalchbrenner et al., 2014; Kim, 2014), recurrent neural networks (Mikolov et al., 2010, 2011; Graves, 2013), and recursive neural networks (Socher et al., 2010, 2011). These generic architectures can be applied to tasks like sentence classification (Zhang et al., 2015; Conneau et al., 2016) and sentence matching (Wan et al., 2016; He et al., 2016), but the model is trained only on data of a particular task.

Recently, Peters et al. (2018) introduce Embeddings from Language Models (ELMo), an approach for learning high-quality, deep contextualized representations using bidirectional language models. With ELMo, they achieve large improvements on six different NLP tasks. Devlin et al. (2018) propose Bidirectional Encoder Representations from Transformers (BERT), a new language representation model that obtains state-of-the-art results on eleven natural language processing tasks. Trained with massive corpora for language modeling, BERT has strong syntactic ability (Goldberg, 2019) and captures generic language features. A typical downstream use of BERT is to fine-tune it for the NLP task at hand. This improves training efficiency, but for inference efficiency, these models are still considerably slower than traditional neural networks.

Model compression. A prominent line of work is devoted to compressing large neural networks to accelerate inference. Early pioneering works include LeCun et al. (1990), who propose a local error-based method for pruning unimportant weights. Recently, Han et al. (2015) propose a simple compression pipeline, achieving 40 times reduction in model size without hurting accuracy. Unfortunately, these techniques induce irregular weight sparsity, which precludes highly optimized computation routines. Thus, others explore pruning entire filters (Li et al., 2016; Liu et al., 2017), with some even targeting device-centric metrics, such as floating-point operations (Tang et al., 2018) and latency (Chen et al., 2018). Still other studies examine quantizing neural networks (Wu et al., 2018); in the extreme, Courbariaux et al. (2016) propose binarized networks with both binary weights and binary activations.

Unlike the aforementioned methods, the knowledge distillation approach (Ba and Caruana, 2014; Hinton et al., 2015) enables the transfer of knowledge from a large model to a smaller, "student" network, which is improved in the process. The student network can use a completely different architecture, since distillation works at the output level. This is important in our case, since our research objective is to study the representation power of shallower neural networks for language understanding, while simultaneously compressing models like BERT; thus, we follow this approach in our work. In the NLP literature, it has previously been used in neural machine translation (Kim and Rush, 2016) and language modeling (Yu et al., 2018).

3 Our Approach

First, we choose the desired teacher and student models for the knowledge distillation approach. Then, we describe our distillation procedure, which comprises two major components: first, the addition of a logits-regression objective, and second, the construction of a transfer dataset, which augments the training set for more effective knowledge transfer.

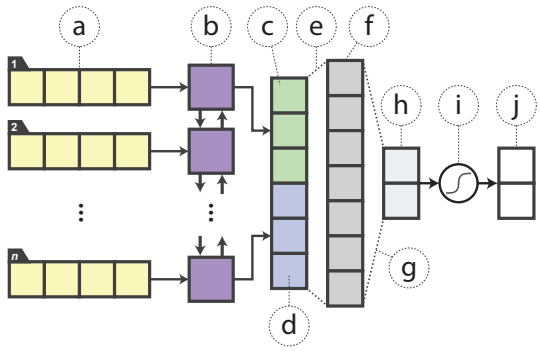

Figure 1: The BiLSTM model for single-sentence classification. The labels are **(a)** input embeddings, **(b)** BiLSTM, **(c, d)** backward and forward hidden states, respectively, **(e, g)** fully-connected layer; **(e)** with ReLU, **(f)** hidden representation, **(h)** logit outputs, **(i)** softmax activation, and **(j)** final probabilities.

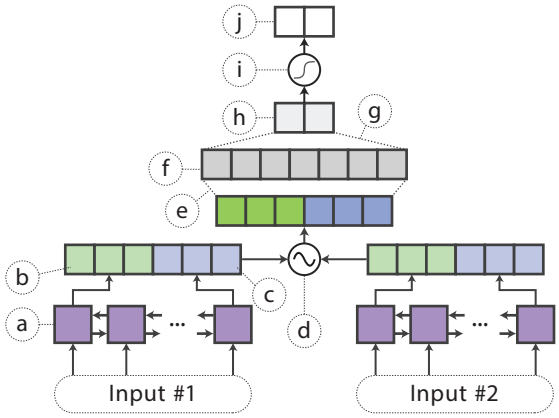

Figure 2: The siamese BiLSTM model for sentence matching, with shared encoder weights for both sentences. The labels are **(a)** BiLSTM, **(b, c)** final backward and forward hidden states, respectively, **(d)** concatenate–compare unit, **(e, g)** fully connected layer; **(e)** with ReLU, **(f)** hidden representation, **(h)** logit outputs, **(i)** softmax activation, and **(j)** final probabilities.

3.1 Model Architecture

For the teacher network, we use the pretrained, fine-tuned BERT (Devlin et al., 2018) model, a deep, bidirectional transformer encoder that achieves state of the art on a variety of language understanding tasks. From an input sentence (pair), BERT computes a feature vector $\boldsymbol{h} \in \mathbb{R}^d$, upon which we build a classifier for the task. For single-sentence classification, we directly build a softmax layer, i.e., the predicted probabilities are $\boldsymbol{y}^{(B)} = \text{softmax}(W\boldsymbol{h})$, where $W \in \mathbb{R}^{k \times d}$ is the softmax weight matrix and k is the number of labels. For sentence-pair tasks, we concatenate the BERT features of both sentences and feed them to a softmax layer. During training, we jointly fine-tune the parameters of BERT and the classifier by maximizing the probability of the correct label, using the cross-entropy loss.

In contrast, our student model is a single-layer BiLSTM with a non-linear classifier. After feeding the input word embeddings into the BiLSTM, the hidden states of the last step in each direction are concatenated and fed to a fully connected layer with rectified linear units (ReLUs), whose output is then passed to a softmax layer for classification (Figure 1). For sentence-pair tasks, we share BiLSTM encoder weights in a siamese architecture between the two sentence encoders, producing sentence vectors $\boldsymbol{h}_{s1}$ and $\boldsymbol{h}_{s2}$ (Figure 2). We then apply a standard concatenate–compare operation (Wang et al., 2018) between the two sentence vectors: $f(\boldsymbol{h}_{s1}, \boldsymbol{h}_{s2}) = [\boldsymbol{h}_{s1}, \boldsymbol{h}_{s2}, \boldsymbol{h}_{s1} \odot \boldsymbol{h}_{s2}, |\boldsymbol{h}_{s1} - \boldsymbol{h}_{s2}|]$, where $\odot$ denotes elementwise multiplication. We feed this output to a ReLU-activated classifier.

It should be emphasized that we restrict the architecture engineering to a minimum to revisit the representation power of BiLSTM itself. We avoid any additional tricks, such as attention and layer normalization.

3.2 Distillation Objective

The distillation approach accomplishes knowledge transfer at the output level; that is, the student network learns to mimic a teacher network's behavior given any data point. In particular, Ba and Caruana (2014) posit that, in addition to a one-hot predicted label, the teacher's predicted probability is also important. In binary sentiment classification, for example, some sentences have a strong sentiment polarity, whereas others appear neutral. If we use only the teacher's predicted one-hot label to train the student, we may lose valuable information about the prediction uncertainty.

The discrete probability output of a neural network is given by

$$\widetilde{y}_i = \text{softmax}(\boldsymbol{z}) = \frac{\exp\{\boldsymbol{w}_i^\top \boldsymbol{h}\}}{\sum_j \exp\{\boldsymbol{w}_j^\top \boldsymbol{h}\}} \quad (1)$$

where $\boldsymbol{w}_i$ denotes the i^{th} row of softmax weight W, and $\boldsymbol{z}$ is equivalent to $\boldsymbol{w}^\top \boldsymbol{h}$. The argument of the softmax function is known as *logits*. Training on logits makes learning easier for the student model since the relationship learned by the teacher

model across all of the targets are equally emphasized (Ba and Caruana, 2014).

The distillation objective is to penalize the mean-squared-error (MSE) loss between the student network's logits against the teacher's logits:

$$\mathcal{L}_{\text{distill}} = ||\boldsymbol{z}^{(B)} - \boldsymbol{z}^{(S)}||_2^2 \quad (2)$$

where $\boldsymbol{z}^{(B)}$ and $\boldsymbol{z}^{(S)}$ are the teacher's and student's logits, respectively. Other measures such as cross entropy with soft targets are viable as well (Hinton et al., 2015); however, in our preliminary experiments, we found MSE to perform slightly better.

At training time, the distilling objective can be used in conjunction with a traditional cross-entropy loss against a one-hot label $\boldsymbol{t}$, given by

$$\begin{aligned}\mathcal{L} &= \alpha \cdot \mathcal{L}_{\text{CE}} + (1-\alpha) \cdot \mathcal{L}_{\text{distill}} \quad (3)\\ &= -\alpha \sum_i t_i \log y_i^{(S)} - (1-\alpha)||\boldsymbol{z}^{(B)} - \boldsymbol{z}^{(S)}||_2^2\end{aligned}$$

When distilling with a labeled dataset, the one-hot target $\boldsymbol{t}$ is simply the ground-truth label. When distilling with an unlabeled dataset, we use the predicted label by the teacher, i.e., $t_i = 1$ if $i = \text{argmax}\, \boldsymbol{y}^{(B)}$ and 0 otherwise.

3.3 Data Augmentation for Distillation

In the distillation approach, a small dataset may not suffice for the teacher model to fully express its knowledge (Ba and Caruana, 2014). Therefore, we augment the training set with a large, unlabeled dataset, with pseudo-labels provided by the teacher, to aid in effective knowledge distillation.

Unfortunately, data augmentation in NLP is usually more difficult than in computer vision. First, there exist a large number of homologous images in computer vision tasks. CIFAR-10, for example, is a subset of the 80 million tiny images dataset (Krizhevsky, 2009). Second, it is possible to synthesize a near-natural image by rotating, adding noise, and other distortions, but if we manually manipulate a natural language sentence, the sentence may not be fluent, and its effect in NLP data augmentation less clear.

In our work, we propose a set of heuristics for task-agnostic data augmentation: we use the original sentences in the small dataset as blueprints, and then modify them with our heuristics, a process analogous to image distortion. Specifically, we randomly perform the following operations.

Masking. With probability p_{mask}, we randomly replace a word with `[MASK]`, which corresponds to an unknown token in our models and the masked word token in BERT. Intuitively, this rule helps to clarify the contribution of each word toward the label, e.g., the teacher network produces less confident logits for "I `[MASK]` the comedy" than for "I loved the comedy."

POS-guided word replacement. With probability p_{pos}, we replace a word with another of the same POS tag. To preserve the original training distribution, the new word is sampled from the unigram word distribution re-normalized by the part-of-speech (POS) tag. This rule perturbs the semantics of each example, e.g., "What do pigs eat?" is different from "How do pigs eat?"

***n*-gram sampling.** With probability p_{ng}, we randomly sample an n-gram from the example, where n is randomly selected from $\{1, 2, \ldots, 5\}$. This rule is conceptually equivalent to dropping out all other words in the example, which is a more aggressive form of masking.

Our data augmentation procedure is as follows: given a training example $\{w_1, \ldots w_n\}$, we iterate over the words, drawing from the uniform distribution $X_i \sim \text{UNIFORM}[0, 1]$ for each w_i. If $X_i < p_{\text{mask}}$, we apply masking to w_i. If $p_{\text{mask}} \leq X_i < p_{\text{mask}} + p_{\text{pos}}$, we apply POS-guided word replacement. We treat masking and POS-guided swapping as mutually exclusive: once one rule is applied, the other is disregarded. After iterating through the words, with probability p_{ng}, we apply n-gram sampling to this entire synthetic example. The final synthetic example is appended to the augmented, unlabeled dataset.

We apply this procedure n_{iter} times per example to generate up to n_{iter} samples from a single example, with any duplicates discarded. For sentence-pair datasets, we cycle through augmenting the first sentence only (holding the second fixed), the second sentence only (holding the first fixed), and both sentences.

4 Experimental Setup

For BERT, we use the large variant $\text{BERT}_{\text{LARGE}}$ (described below) as the teacher network, starting with the pretrained weights and following the original, task-specific fine-tuning procedure (Devlin et al., 2018). We fine-tune four models using the Adam optimizer with learning rates $\{2, 3, 4, 5\} \times 10^{-5}$, picking the best model on the validation set. We avoid data augmentation during fine-tuning.

For our models, we feed the original dataset together with the synthesized examples to the task-specific, fine-tuned BERT model to obtain the predicted logits. We denote our distilled BiLSTM trained on soft logit targets as $\text{BiLSTM}_{\text{SOFT}}$, which corresponds to choosing $\alpha = 0$ in Section 3.2. Preliminary experiments suggest that using only the distillation objective works best.

4.1 Datasets

We conduct experiments on the General Language Understanding Evaluation (GLUE; Wang et al., 2018) benchmark, a collection of six natural language understanding tasks that are classified into three categories: single-sentence tasks, similarity and paraphrase tasks, and inference tasks. Due to restrictions in time and computational resources, we choose the most widely used dataset from each category, as detailed below.

SST-2. Stanford Sentiment Treebank 2 (SST-2; Socher et al., 2013) comprises single sentences extracted from movie reviews for binary sentiment classification (positive vs. negative). Following GLUE, we consider sentence-level sentiment only, ignoring the sentiment labels of phrases provided by the original dataset.

MNLI. The Multi-genre Natural Language Inference (MNLI; Williams et al., 2017) corpus is a large-scale, crowdsourced entailment classification dataset. The objective is to predict the relationship between a pair of sentences as one of entailment, neutrality, or contradiction. MNLI-m uses development and test sets that contain the same genres from the training set, while MNLI-mm represents development and test sets from the remaining, mismatched genres.

QQP. Quora Question Pairs (QQP; Shankar Iyer and Csernai, 2017) consists of pairs of potentially duplicate questions collected from Quora, a question-and-answer website. The binary label of each question pair indicates redundancy.

4.2 Hyperparameters

We choose either 150 or 300 hidden units for the BiLSTM, and 200 or 400 units in the ReLU-activated hidden layer, depending on the validation set performance. Following Kim (2014), we use the traditional 300-dimensional word2vec embeddings trained on Google News and multi-channel embeddings. For optimization, we use AdaDelta (Zeiler, 2012) with its default learning rate of 1.0 and $\rho = 0.95$. For SST-2, we use a batch size of 50; for MNLI and QQP, due to their larger size, we choose 256 for the batch size.

For our dataset augmentation hyperparameters, we fix $p_{\text{mask}} = p_{\text{pos}} = 0.1$ and $p_{\text{ng}} = 0.25$ across all datasets. These values have *not* been tuned at all on the datasets—these are the first values we chose. We choose $n_{\text{iter}} = 20$ for SST-2 and $n_{\text{iter}} = 10$ for both MNLI and QQP, since they are larger.

4.3 Baseline Models

BERT (Devlin et al., 2018) is a multi-layer, bidirectional transformer encoder that comes in two variants: $\text{BERT}_{\text{BASE}}$ and the larger $\text{BERT}_{\text{LARGE}}$. $\text{BERT}_{\text{BASE}}$ comprises 12 layers, 768 hidden units, 12 self-attention heads, and 110M parameters. $\text{BERT}_{\text{LARGE}}$ uses 24 layers, 1024 hidden units, 16 self-attention heads, and 340M parameters.

OpenAI GPT (Radford et al., 2018) is, like BERT, a generative pretrained transformer (GPT) encoder fine-tuned on downstream tasks. Unlike BERT, however, GPT is unidirectional and only makes use of previous context at each time step.

GLUE ELMo baselines. In the GLUE paper, Wang et al. (2018) provide a BiLSTM-based model baseline trained on top of ELMo and jointly fine-tuned across *all* tasks. This model contains 4096 units in the ELMo BiLSTM and more than 93 million total parameters. In the BERT paper, Devlin et al. (2018) provide the same model but a result slightly different from Wang et al. (2018). For fair comparison, we report both results.

5 Results and Discussion

We present the results of our models as well as baselines in Table 1. For QQP, we report both F_1 and accuracy, since the dataset is slightly unbalanced. Following GLUE, we report the average score of each model on the datasets.

5.1 Model Quality

To verify the correctness of our implementation, we train the base BiLSTM model on the original labels, without using distillation (row 7). Across all three datasets, we achieve scores comparable with BiLSTMs from previous works (rows 8 and 9), suggesting that our implementation is fair. Note that, on MNLI, the two baselines differ by 4% in accuracy (rows 8 and 9). None of the non-distilled BiLSTM baselines outperform BERT's

#	Model	SST-2	QQP	MNLI-m	MNLI-mm
		Acc	F_1/Acc	Acc	Acc
1	$\text{BERT}_{\text{LARGE}}$ (Devlin et al., 2018)	94.9	72.1/89.3	86.7	85.9
2	$\text{BERT}_{\text{BASE}}$ (Devlin et al., 2018)	93.5	71.2/89.2	84.6	83.4
3	OpenAI GPT (Radford et al., 2018)	91.3	70.3/88.5	82.1	81.4
4	BERT ELMo baseline (Devlin et al., 2018)	90.4	64.8/84.7	76.4	76.1
5	GLUE ELMo baseline (Wang et al., 2018)	90.4	63.1/84.3	74.1	74.5
6	Distilled $\text{BiLSTM}_{\text{SOFT}}$	**90.7**	**68.2/88.1**	**73.0**	**72.6**
7	BiLSTM (our implementation)	86.7	63.7/86.2	68.7	68.3
8	BiLSTM (reported by GLUE)	85.9	61.4/81.7	70.3	70.8
9	BiLSTM (reported by other papers)	87.6†	– /82.6‡	66.9*	66.9*

Table 1: Test results on different datasets. The BiLSTM results reported by other papers are drawn from Zhou et al. (2016),† Wang et al. (2017),‡ and Williams et al. (2017).* All of our test results are obtained from the GLUE benchmark website.

ELMo baseline (row 4)—our implementation, although attaining a higher accuracy for QQP, falls short in F_1 score.

We apply our distillation approach of matching logits using the augmented training dataset, and achieve an absolute improvement of 1.9–4.5 points against our base BiLSTM. On SST-2 and QQP, we outperform the best reported ELMo model (row 4), coming close to GPT. On MNLI, our results trail ELMo's by a few points; however, they still represent a 4.3-point improvement against our BiLSTM, and a 1.8–2.7-point increase over the previous best BiLSTM (row 8). Overall, our distilled model is competitive with two previous implementations of ELMo BiLSTMs (rows 4–5), suggesting that shallow BiLSTMs have greater representation power than previously thought.

We do not, however, outperform the deep transformer models (rows 1–3), doing 4–7 points worse, on average. Nevertheless, our model has much fewer parameters and better efficiency, as detailed in the following section.

5.2 Inference Efficiency

For our inference speed and parameter analysis, we use the open-source PyTorch implementations for BERT[2] and ELMo (Gardner et al., 2017). On a single NVIDIA V100 GPU, we perform model inference with a batch size of 512 on all 67350 sentences of the SST-2 training set. As shown in Table 2, our single-sentence model uses 98 and 349 times fewer parameters than ELMo and $\text{BERT}_{\text{LARGE}}$, respectively, and is 15 and 434 times faster. At 2.2 million parameters, the variant with 300-dimensional LSTM units is twice as large, though still substantially smaller than ELMo. For sentence-pair tasks, the siamese counterpart uses no pairwise word interactions, unlike previous state of the art (He and Lin, 2016); its runtime thus scales linearly with sentence length.

[2] https://goo.gl/iRPhjP

	# of Par.	Inference Time
$\text{BERT}_{\text{LARGE}}$	335 (349×)	1060 (434×)
ELMo	93.6 (98×)	36.71 (15×)
$\text{BiLSTM}_{\text{SOFT}}$	0.96 (1×)	2.44 (1×)

Table 2: Single-sentence model size and inference speed on SST-2. # of Par. denotes number of millions of parameters, and inference time is in seconds.

6 Conclusion and Future Work

In this paper, we explore distilling the knowledge from BERT into a simple BiLSTM-based model. The distilled model achieves comparable results with ELMo, while using much fewer parameters and less inference time. Our results suggest that shallow BiLSTMs are more expressive for natural language tasks than previously thought.

One direction of future work is to explore extremely simple architectures in the extreme, such as convolutional neural networks and even support vector machines and logistic regression. Another opposite direction is to explore slightly more complicated architectures using tricks like pairwise word interaction and attention.

Acknowledgements

This research was enabled in part by resources provided by Compute Ontario and Compute Canada. This research was also supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

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.

Changan Chen, Frederick Tung, Naveen Vedula, and Greg Mori. 2018. Constraint-aware deep neural network compression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 400–415.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for text classification. *arXiv:1606.01781*.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv:1602.02830*.

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

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform. *arXiv:1803.07640*.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *arXiv:1901.05287*.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv:1308.0850*.

Song Han. 2017. Efficient methods and hardware for deep learning.

Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv:1510.00149*.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948.

Hua He, John Wieting, Kevin Gimpel, Jinfeng Rao, and Jimmy Lin. 2016. UMD-TTIC-UW at SemEval-2016 task 1: Attention-based multi-perspective convolutional neural networks for textual similarity measurement. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1103–1108.

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

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv:1404.2188*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Yann LeCun, John S. Denker, and Sara A. Solla. 1990. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605.

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

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2736–2744.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531.

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)*, volume 1, pages 2227–2237.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. *arXiv:1603.06807*.

Nikhil Dandekar Shankar Iyer and Kornl Csernai. 2017. First Quora dataset release: Question pairs.

Richard Socher, Cliff C. Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 129–136.

Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, volume 2010, pages 1–9.

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.

Raphael Tang, Ashutosh Adhikari, and Jimmy Lin. 2018. FLOPs as a direct optimization objective for learning sparse neural networks. *arXiv:1811.03060*.

Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv:1804.07461*.

Dingquan Wang and Jason Eisner. 2016. The galactic dependencies treebanks: Getting more data by synthesizing new languages. *Transactions of the Association for Computational Linguistics*, 4:491–505.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4144–4150.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv:1704.05426*.

Shuang Wu, Guoqi Li, Feng Chen, and Luping Shi. 2018. Training and inference with integers in deep neural networks. In *International Conference on Learning Representations*.

Seunghak Yu, Nilesh Kulkarni, Haejun Lee, and Jihie Kim. 2018. On-device neural language model based word prediction. *Proceedings of COLING 2018, the 28th International Conference on Computational Linguistics: Technical Papers*, page 128.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv:1212.5701*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3485–3495.