Natural Language Generation for Effective Knowledge Distillation

Raphael Tang, Yao Lu, and **Jimmy Lin**
David R. Cheriton School of Computer Science
University of Waterloo

Abstract

Knowledge distillation can effectively transfer knowledge from BERT, a deep language representation model, to traditional, shallow word embedding-based neural networks, helping them approach or exceed the quality of other heavyweight language representation models. As shown in previous work, critical to this distillation procedure is the construction of an unlabeled transfer dataset, which enables effective knowledge transfer. To create transfer set examples, we propose to sample from pretrained language models fine-tuned on task-specific text. Unlike previous techniques, this directly captures the purpose of the transfer set. We hypothesize that this principled, general approach outperforms rule-based techniques. On four datasets in sentiment classification, sentence similarity, and linguistic acceptability, we show that our approach improves upon previous methods. We outperform OpenAI GPT, a deep pretrained transformer, on three of the datasets, while using a single-layer bidirectional LSTM that runs at least ten times faster.

1 Introduction

That bigger neural networks plus more data equals higher quality is a tried-and-true formula. In the natural language processing (NLP) literature, the recent darling of this mantra is the deep, pretrained language representation model. After pretraining hundreds of millions of parameters on vast amounts of text, models such as BERT (Bidirectional Encoder Representations from Transformers; Devlin et al., 2018) achieve remarkable state of the art in question answering, sentiment analysis, and sentence similarity tasks, to list a few.

Does this progress mean, then, that classic, shallow word embedding-based neural networks are noncompetitive? Not quite. Recently, Tang et al. (2019) demonstrate that knowledge distillation (Ba and Caruana, 2014; Hinton et al., 2015) can transfer knowledge from BERT to small, traditional neural networks, helping them approach or exceed the quality of much larger pretrained long short-term memory (LSTM; Hochreiter and Schmidhuber, 1997) language models, such as ELMo (Embeddings from Language Models; Peters et al., 2018).

As shown in Tang et al. (2019), crucial to knowledge distillation is constructing a transfer dataset of unlabeled examples. In this paper, we explore how to construct such an effective transfer set. Previous approaches comprise manual data curation, a meticulous method where the end user manually selects a corpus similar enough to the present task, and rule-based techniques, where a transfer set is fabricated from the training set using a set of data augmentation rules. However, these rules only indirectly model the purpose of the transfer set, which is to provide more input drawn from the task-specific data distribution. Hence, we instead propose to construct the transfer set by generating text with pretrained language models fine-tuned on task-specific text. We validate our approach on four small- to mid-sized datasets in sentiment classification, sentence similarity, and linguistic acceptability.

We claim two contributions: first, we elucidate a novel approach for constructing the transfer set in knowledge distillation. Second, we are the first to outperform OpenAI GPT (Radford et al., 2018) in sentiment classification and sentence similarity with a single-layer bidirectional LSTM (BiLSTM) that runs more than ten times faster, *without* pretraining or domain-specific data curation. We make our datasets and codebase public in a GitHub repository.[1]

[1] `https://github.com/castorini/d-bert`

2 Background and Related Work

Ba and Caruana (2014) propose knowledge distillation, a method for improving the quality of a smaller *student* model by encouraging it to match the outputs of a larger, higher-quality *teacher* network. Concretely, suppose $h_S(\cdot)$ and $h_T(\cdot)$ respectively denote the untrained student and trained teacher models, and we are given a training set of inputs $\mathcal{S} = \{x_1, \dots, x_N\}$. On classification tasks, the model outputs are log probabilities; on regression tasks, the outputs are as-is. Then, the distillation objective $\mathcal{L}_{\text{KD}}$ is

$$\mathcal{L}_{\text{KD}} = \frac{1}{N} \sum_{i=1}^{N} \|h_S(x_i) - h_T(x_i)\|_2^2 \qquad (1)$$

Hinton et al. (2015) alternatively use Kullback–Leibler divergence for classification, along with additional hyperparameters. For simplicity and generality, we stick with the original mean-squared error (MSE) formulation. We minimize $\mathcal{L}_{\text{KD}}$ end-to-end with backpropagation, updating the student's parameters and fixing the teacher's. $\mathcal{L}_{\text{KD}}$ can optionally be combined with the original, supervised cross-entropy or MSE loss; following Tang et al. (2019) and Shi et al. (2019), we optimize only $\mathcal{L}_{\text{KD}}$ for training the student.

Using only the given training set for $\mathcal{S}$, however, is often insufficient. Thus, Ba and Caruana (2014) augment $\mathcal{S}$ with a *transfer* set comprising unlabeled input, providing the student with more examples to distill from the teacher. Techniques for constructing this transfer set consist of either manual data curation or unprincipled data synthesis rules. Ba and Caruana (2014) choose images from the 80 million tiny images dataset, which is a superset of their dataset. In the NLP domain, Tang et al. (2019) propose text perturbation rules for creating a transfer set from the training set, achieving results comparable to ELMo using a BiLSTM with 100 times fewer parameters.

We wish to avoid these previous approaches. Manual data curation requires the researcher to select an unlabeled set similar enough to the target dataset, a difficult-to-impossible task for many datasets in, for example, linguistic acceptability and sentence similarity. Rule-based techniques, while general, unfortunately deviate from the *true* purpose of modeling the input distribution; hence, we hypothesize that they are less effective than a principled approach, which we detail below.

3 Our Approach

In knowledge distillation, the student perceives the oracular teacher to be the true $p(Y|X)$, where X and Y respectively denote the input sentence and label. This is reasonable, since the student treats the teacher output y as ground truth, given some sentence x comprising words $\{w_1, \dots, w_n\}$. The purpose of the transfer set is, then, to provide additional input sentences for querying the teacher. To construct such a set, we propose the following: first, we parameterize $p(X)$ directly as a language model $p(w_1, \dots, w_n) = \Pi_{i=1}^{n} p(w_i|w_1, \dots, w_{i-1})$ trained on the given sentences $\{x_1, \dots, x_N\}$. Then, to generate unlabeled examples, we sample from the language model, i.e., the i^{th} word of a sentence is drawn from $p(w_i|w_1, \dots, w_{i-1})$. We stop upon generating the special end-of-sentence token `[EOS]`, which we append to each sentence while fine-tuning the language model (LM).

Unlike previous methods, our approach directly parameterizes $p(X)$ to provide unlabeled examples. We hypothesize that this approach outperforms ad hoc rule-based methods, which only indirectly model the input distribution $p(X)$.

Sentence-pair modeling. To language model sentence pairs, we follow Devlin et al. (2018) and join both sentences with a special separator token `[SEP]` between, treating the resulting sequence as a single contiguous sentence.

3.1 Model Architecture

For simplicity and efficient inference, our student models use the same single-layer BiLSTM models from Tang et al. (2019)—see Figures 1 and 2.

First, we map an input sequence of words to their corresponding word2vec embeddings, trained on Google News. Next, for single-sentence tasks, these embeddings are fed into a single-layer BiLSTM encoder to yield concatenated forward and backward states $\mathbf{h} = [\mathbf{h}_f; \mathbf{h}_b]$. For sentence-pair tasks, we encode each sentence separately using a BiLSTM to yield $\mathbf{h}_1$ and $\mathbf{h}_2$. To produce a single vector $\mathbf{h}$, following Wang et al. (2018), we compute $\mathbf{h} = [\mathbf{h}_1; \mathbf{h}_2; \delta(\mathbf{h}_1, \mathbf{h}_2); \mathbf{h}_1 \cdot \mathbf{h}_2]$, where $\cdot$ denotes elementwise multiplication and δ denotes elementwise absolute difference. Finally, for both single- and paired-sentence tasks, $\mathbf{h}$ is passed through a multilayer perceptron (MLP) with one hidden layer that uses a rectified linear unit (ReLU) activation. For classification, the fi-

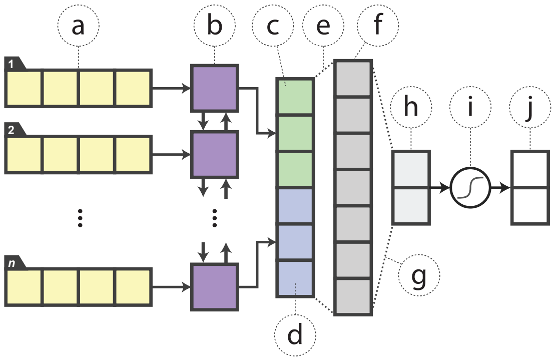

Figure 1: Illustration of the single-sentence BiLSTM, copied from Tang et al. (2019). The labels are as follows: (a) word embeddings (b) BiLSTM layer (c) final forward hidden state (d) final backward hidden state (e) nonlinear layer (f) the final representation (g) fully-connected layer (h) logits or similarity score (i) softmax activation for classification tasks; identity for regression (j) final probabilities or score.

nal output is interpreted as the logits of each class; for real-valued sentence similarity, the final output is a single score.

Our teacher model is the large variant of BERT, a deep pretrained language representation model that achieves close to state of the art (SOTA) on our tasks. Extremely recent, improved pretrained models like XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019) likely offer greater benefits to the student model, but BERT is widely used and sufficient for the point of this paper. We follow the same experimental procedure in Devlin et al. (2018) and fine-tune BERT end-to-end for each task, varying only the final classifier layer for the desired number of classes.

Language modeling. For creating the transfer set, we apply two public, state-of-the-art language models: the word-level Transformer-XL (TXL; Dai et al., 2019) pretrained on WikiText-103 (Merity et al., 2017), which is derived from Wikipedia, and the subword-level GPT-2 (345M version; Radford et al., 2019) pretrained on WebText, which represents a large web corpus that *excludes* Wikipedia. Other models exist, but we choose these two since they represent the state of the art. We name the GPT-2 and TXL-constructed transfer sets $\text{TS}_{\text{GPT-2}}$ and TS_{TXL}, respectively.

4 Experimental Setup

We validate our approach on four datasets in sentiment classification, linguistic acceptability, sentence similarity, and paraphrasing: Stanford Sentiment Treebank-2 (SST-2; Socher et al., 2013), the Corpus of Linguistic Acceptability (CoLA; Warstadt et al., 2018), Semantic Textual Similarity Benchmark (STS-B; Cer et al., 2017), and Microsoft Research Paraphrase Corpus (MRPC; Dolan and Brockett, 2005). SST-2 is a binary polarity dataset of single-sentence movie reviews. CoLA is a single-sentence grammaticality task, with expertly annotated binary judgements. STS-B comprises sentence pairs labeled with real-valued similarity between 1 and 5. Lastly, MRPC has sentence pairs with binary labels denoting semantic equivalence. We pick these four tasks from the General Language Understanding Evaluation (GLUE; Wang et al., 2018) benchmark, and submit results to their public evaluation server.[2]

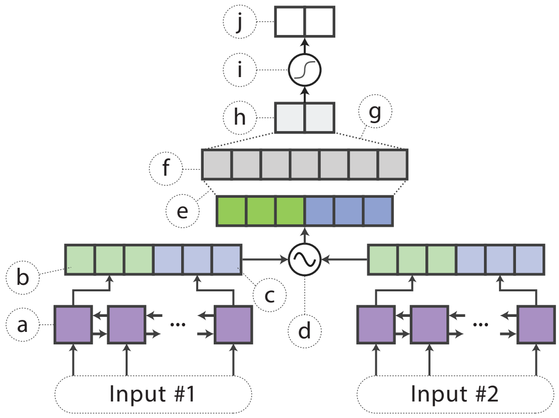

Figure 2: Illustration of the sentence-pair BiLSTM, copied from Tang et al. (2019). The labels are as follows: (a) BiLSTM layer (b) final forward hidden state (c) final backward hidden state (d) comparison unit, as detailed in the text (e) nonlinear layer (f) the final representation (g) fully-connected layer (h) logits or similarity score (i) softmax activation for classification tasks; identity for regression (j) final probabilities or score.

4.1 Baselines

As a sanity check, we attempt knowledge distillation *without* a transfer set, as well as training our BiLSTM from scratch on the original labels. We compare to the best official GLUE test results reported for single- and multi-task ELMo models, OpenAI GPT, single- and multi-task single-layer BiLSTMs, and the SOTA before GPT. ELMo and GPT are pretrained language representation models with around a hundred million parameters. We name our distilled model $\text{BiLSTM}_{\text{KD}}$.

[2] http://gluebenchmark.com

Transfer set construction baselines. For our rule-based baseline, we use the masking and part of speech (POS)-guided word swapping rules as originally suggested by Tang et al. (2019), which consist of the following: iterating through a dataset's sentences, we replace 10% of the words with the masking token `[MASK]`. We swap another mutually exclusive 10% of the words with others of the same POS tag from the vocabulary, randomly sampling by unigram probability. For sentence-pair tasks, we apply the rules to the first sentence only, then the second only, and, finally, both. Discarding any duplicates, we repeat this entire process until meeting the target number of transfer set sentences. Tang et al. (2019) also suggest to sample n-grams; however, we omit this rule, since our preliminary experiments find that it hurts accuracy. We call this method TS_{MP}.

For our unlabeled dataset baseline, we choose the document-level IMDb movie reviews dataset (Diao et al., 2014) as our transfer set for SST-2. To match the single-sentence SST-2, we break paragraphs into individual linguistic sentences and, hence, multiple transfer set examples. To confirm that this is domain sensitive, we also apply it to the out-of-domain CoLA task in linguistic acceptability. We are unable to find a suitable unlabeled set for our other tasks—by construction, most sentence-pair datasets require manual balancing to prevent an overabundance of a single class, e.g., dissimilar examples in sentence similarity. We call this method TS_{IMDb}.

4.2 Training and Hyperparameters

We fine-tune our pretrained language models using largely the same procedure from Devlin et al. (2018). For fair comparison, we use 800K sentences for *all* transfer sets, including TS_{IMDb}. For our BiLSTM student models, we follow Tang et al. (2019) and use ADADELTA (Zeiler, 2012) with its default LR of 1.0 and $\rho = 0.95$. We train our models for 30 epochs, choosing the best performing on the standard development set. As is standard, for classification tasks, we minimize the negative log-likelihood; for regression, the mean-squared error. Depending on the loss on the development set, we choose either 150 or 300 LSTM units, and 200 or 400 hidden MLP units. This results in a model size between 1–3 million parameters. We use the 300-dimensional word2vec vectors trained on Google News, initializing out-of-vocabulary (OOV) vectors from $\text{UNIFORM}[-0.25, 0.25]$, following Kim (2014), along with multichannel embeddings.

To fine-tune our pretrained language models, we use Adam (Kingma and Ba, 2014) with a learning rate (LR) linear warmup proportion of 0.1, linearly decaying the LR afterwards. We choose a batch size of eight and one fine-tuning epoch, which is sufficient for convergence. We tune the LR from $\{1, 5\} \times 10^{-5}$ based on word-level perplexity on the development set.

5 Results and Discussion

We present our results in Table 1. As an initial sanity check, we confirm that our BiLSTM (row 11) is acceptably similar to the previous best reported BiLSTM (row 5). We also verify that a transfer set is necessary—see rows 10 and 11, where using only the training dataset for distillation is insufficient. We further confirm that TS_{IMDb} works poorly for the out-of-domain CoLA dataset (row 8). Note that the absolute best result on SST-2 before BERT is 93.2, from Radford et al. (2017), but that approach demands copious amounts of domain-specific data from the practitioner.

5.1 Quality and Efficiency

Of the transfer set construction approaches, our principled generation methods consistently achieve the highest results (see Table 1, rows 6 and 7), followed by the rule-based TS_{MP} and the manually curated TS_{IMDb} (rows 8 and 9). $\text{TS}_{\text{GPT-2}}$ is especially effective for CoLA, yielding a respective 12.5- and 30-point increase in Matthew's Correlation Coefficient (MCC) over TS_{MP} and training from scratch.

Interestingly, on SST-2, the synthetic GPT-2 samples outperform handwritten movie reviews from IMDb. Unlike the rule-based TS_{MP}, our LM-driven approaches outperform ELMo on all four tasks. $\text{TS}_{\text{GPT-2}}$, our best method, reaches GPT parity on all but CoLA, establishing domain-agnostic, pre-BERT SOTA on SST-2 and STS-B.

Our models use between one and three million parameters, which is at least 30 and 40 times smaller than ELMo and GPT, respectively. This represents an improvement over the previous SOTA—see the official GLUE leaderboard and Devlin et al. (2018) for specifics.

It should be emphasized that using fewer model parameters does *not* necessarily reduce the total

#	Model	SST-2	CoLA	STS-B	MRPC
		Acc.	MCC	r/ρ	F$_1$/Acc.
1	BERT$_{large}$ (Devlin et al., 2018)	94.9	60.5	86.5/87.6	89.3/85.4
2	OpenAI GPT (Radford et al., 2018)	91.3	**45.4**	82.0/80.0	82.3/75.7
3	Pre-OpenAI SOTA (Devlin et al., 2018)	90.2†	35.0	81.0/–	**86.0/80.4**
4	ELMo BiLSTM (Wang et al., 2018)	90.4	36.0	74.2/72.3	84.9/78.0
5	BiLSTM scratch (GLUE leaderboard)	85.9	15.7	70.3/67.8	81.8/74.3
6	BiLSTM$_{\text{KD}}$+TS$_{\text{GPT-2}}$	**92.7**	40.0	**82.1/80.7**	85.5/80.2
7	BiLSTM$_{\text{KD}}$+TS$_{\text{TXL}}$	91.9	36.5	82.0/80.4	85.1/79.3
8	BiLSTM$_{\text{KD}}$+TS$_{\text{IMDb}}$	92.0	18.8	–	–
9	BiLSTM$_{\text{KD}}$+TS$_{\text{MP}}$	90.7	27.5	81.1/79.3	82.4/76.1
10	BiLSTM$_{\text{KD}}$ (no TS)	88.4	0.0	68.2/65.8	78.0/69.7
11	BiLSTM scratch (ours)	87.6	9.5	66.9/64.3	80.9/69.4

Table 1: GLUE test results for our models, along with previous comparison points. Bolded are the best scores from rows 2–11. †For fair comparison, this result is copied from Looks et al. (2017), which represents the best *domain-agnostic* approach; the rest in row 3 is from Devlin et al. (2018) and the GLUE website.

#	Dataset	SST-2		CoLA		STS-B	MRPC	
		U3$_{\%}$	p/n	U3$_{\%}$	p/n	U3$_{\%}$	U3$_{\%}$	p/n
1	TS$_{\text{GPT-2}}$	77%	1.14	88%	2.71	83%	82%	0.41
2	TS$_{\text{TXL}}$	76%	1.29	87%	1.51	80%	82%	0.25
3	TS$_{\text{IMDb}}$	65%	1.65	65%	8.35	–	–	–
4	TS$_{\text{MP}}$	44%	1.23	69%	1.10	62%	60%	1.38
5	Training	20%	1.26	64%	2.38	66%	64%	2.07

Table 2: Diversity and generation statistics.

Model	SST-2			CoLA			STS-B			MRPC		
	OOV	ppl	bpc	OOV	ppl	bpc	OOV	ppl	bpc	OOV	ppl	bpc
GPT-2	0%	67	1.3	0%	60	1.1	0%	35	1.2	0%	19	1.3
TXL	2.9%	77	1.8	0.1%	32	1.2	1.4%	32	1.9	1.0%	17	2.5

Table 3: Language modeling statistics.

disk usage. All traditional, word embedding-based models require storing the word vectors, which obviously precludes many on-device applications. Instead, the main benefit is that these shallow BiLSTMs perform inference an order of magnitude faster than GPT, which is mostly important for server-based, in-production NLP systems.

5.2 Language Generation Analysis

To characterize the transfer sets, we present diversity statistics in Table 2. U3$_{\%}$ denotes the average percentage of unique trigrams (Fedus et al., 2018) across sequential dataset chunks of size M, where M matches the original dataset size for fairness. Specifically, it represents the following:

$$\frac{1}{K}\sum_{i=1}^{K}\frac{\text{\# unique trigrams in } x_{((i-1)M+1):iM}}{\text{\# total trigrams in } x_{((i-1)M+1):iM}} \qquad (2)$$

where $K = \lfloor N/M \rfloor$ and $\{x_1, \ldots, x_N\}$ the dataset. We find that TS$_{\text{GPT-2}}$ and TS$_{\text{TXL}}$ (rows 1 and 2) contain more unique trigrams than TS$_{\text{MP}}$, the original training set, and, surprisingly, handwritten movie reviews from IMDb (see rows 3–5).

To examine whether the class distribution of the transfer sets matches the original, we compute p/n, the positive-to-negative label ratio. Based on the statistics, we conclude that p/n varies wildly among the methods and datasets, with our LM-generated transfer sets differing substantially on MRPC, e.g., TS$_{\text{GPT-2}}$'s 0.41 versus the original's 2.07. This suggests that similar examples are more difficult to generate than dissimilar ones.

Finally, to characterize the LMs, we report GPT-2's and TXL's word-level perplexity (PPL) and bits per character (BPC) on the development sets, as well as the percentage of OOV tokens on the dataset—see Table 3, where lower scores are better. GPT-2 has practically no OOV for English, due to its byte-pair encoding scheme. In spite of using half as many parameters, GPT-2 is better at character-level language modeling than TXL is on all datasets, and its word-level PPL is similar, except on CoLA. As a rough analysis, BPC is a stronger predictor of improved quality than PPL is. Across the datasets, distillation quality strictly increases with decreasing BPC, unlike PPL, suggesting that character-level modeling is more important for constructing an effective transfer set.

Set	Example
TS$_{\text{GPT-2}}$	cansfield 's further oeuvre encompasses it somehow , and the surreal feels natural . `[EOS]`
TS$_{\text{TXL}}$	ethereal and plot of irony and irony and , most importantly , subtle suspense and spirit game - of - humor . `[EOS]`
TS$_{\text{MP}}$	what should have been a cutting hollywood satire is `[MASK]` about as fresh as last week 's issue of variety . `[EOS]`
TS$_{\text{IMDb}}$	but it the end, the film is a big steaming pile of...y'know. `[EOS]`
Training	the cinematography to the outstanding soundtrack and unconventional narrative `[EOS]`

Table 4: Generation examples on SST-2.

Generation examples. We present a random example from each transfer set in Table 4 for SST-2. The generated samples ostensibly consist of movie reviews and contain acceptable linguistic structure, despite only one epoch of fine-tuning. Due to space limitations, we show only SST-2; however, the other transfer sets are public for examination in our GitHub repository.

6 Conclusions and Future Work

We propose using text generation for constructing the transfer set in knowledge distillation. We validate our hypothesis that generating text using pretrained LMs outperforms manual data curation and rule-based techniques: the former in generality, and the latter efficacy. Across multiple datasets, we achieve OpenAI GPT-level quality using a single-layer BiLSTM.

The presented techniques can be readily extended to sequence-to-sequence-level knowledge distillation for applications in neural machine translation and logical form induction. Another line of future work involves applying the techniques to knowledge distillation for traditional, in-production NLP systems.

Acknowledgments

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

References

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Zihang Dai, Zhilin Yang, Yiming Yang, William W. Cohen, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. *arXiv:1901.02860*.

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

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

William Fedus, Ian Goodfellow, and Andrew M. Dai. 2018. MaskGAN: Better text generation via filling in the _______. In *International Conference on Learning Representations*.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692*.

Moshe Looks, Marcello Herreshoff, DeLesley Hutchins, and Peter Norvig. 2017. Deep learning with dynamic computation graphs. In *International Conference on Learning Representations*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

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)*.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv:1704.01444*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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

Yangyang Shi, Mei-Yuh Hwang, Xin Lei, and Haoyu Sheng. 2019. Knowledge distillation for recurrent neural network language modeling with trust regularization. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

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*.

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:1903.12136*.

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*.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2018. Neural network acceptability judgments. *arXiv:1805.12471*.

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

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