Exploring the Limits of Simple Learners in Knowledge Distillation for Document Classification with DocBERT

Ashutosh Adhikari[†], **Achyudh Ram**[†], **Raphael Tang**[†],
William L. Hamilton[‡], **Jimmy Lin**[†]
[†]David R. Cheriton School of Computer Science, University of Waterloo
[‡] Mila, McGill University
{adadhika, arkeshav, r33tang, jimmylin}@uwaterloo.ca
wlh@cs.mcgill.ca

Abstract

Fine-tuned variants of BERT are able to achieve state-of-the-art accuracy on many natural language processing tasks, although at significant computational costs. In this paper, we verify BERT's effectiveness for document classification and investigate the extent to which BERT-level effectiveness can be obtained by different baselines, combined with knowledge distillation—a popular model compression method. The results show that BERT-level effectiveness can be achieved by a single-layer LSTM with at least 40× fewer FLOPS and only ∼3% parameters. More importantly, this study analyzes the limits of knowledge distillation as we distill BERT's knowledge all the way down to linear models—a relevant baseline for the task. We report substantial improvement in effectiveness for even the simplest models, as they capture the knowledge learnt by BERT.

1 Introduction

Transformer-based (Vaswani et al., 2017) pre-trained contextual word embedding models such as BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019) currently power many of the state-of-the-art models across various natural language processing (NLP) tasks. However, these models consume immense computational resources (Strubell et al., 2019). With the surge of such pre-trained models being developed in quick succession, there is a need for effective compression techniques for their inexpensive deployment.

Knowledge distillation (KD; Hinton et al. 2015; Ba and Caruana 2014) has been shown to be a fairly straightforward and effective model-agnostic compression method, which transfers knowledge learnt by huge models into more efficient models. In this work, we investigate if BERT-level effectiveness can be achieved by more efficient models using KD. And more importantly, if so, *how simple* can these models be?

We investigate these questions through the lens of document classification—a setting where these computational concerns are particularly relevant due to potentially long document lengths. Further, in previous work, neural networks as an architectural choice have been questioned owing to the effectiveness of simple bag-of-words baselines (Adhikari et al., 2019).

We first confirm that a fine-tuned BERT model leads to state-of-the-art model quality by a substantial margin on standard document classification benchmarks. Following this, we investigate the extent to which BERT-level effectiveness can be obtained by various different baselines, combined with KD. We demonstrate, quite surprisingly, that it is possible to apply KD successfully on impoverished student models, such as a single-layer convolutional neural network (CNN) (Kim, 2014) and even linear models. The key contributions of this work are as follows:

1. We develop and release[*] a fine-tuned BERT model (DocBERT), which achieves state-of-the-art model quality for document classification. While this finding is perhaps obvious, we carefully document experimental results.

2. We explore the limits of KD from BERT by distilling to substantially simpler and more efficient baselines than previous work (e.g., logistic regression). We are the first, to our knowledge, to demonstrate the working of KD all the way down to linear models.

3. We show that an LSTM baseline (40× faster than BERT$_{base}$), combined with KD can achieve BERT-level model quality.

[*] https://github.com/castorini/hedwig

Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP-2020), pages 72–77
July 9, 2020. ©2020 Association for Computational Linguistics

2 Background and Methods

Typically, the task of document classification deals with classifying long texts (documents). More often than not, a document may be associated with more than one label, thus exposing the classifiers to multi-label classification and class imbalance. Here, we review a subset of approaches developed to solve the task and highlight the methods that we compare and build upon in this work.

2.1 Document Classification Models

Neural network-based models. In recent years neural network-based architectures have dominated the task of document classification. Many researchers (Kim, 2014; Conneau et al., 2017; Johnson and Zhang, 2017) show convolutional neural networks to be effective for classifying single-label short texts. Furthermore, Liu et al. (2017) develop a variant of the popular KimCNN (Kim, 2014), XML-CNN, for addressing the multi-label nature of document classification, which they call extreme classification. Alternatively, others (Yang et al., 2016; Adhikari et al., 2019; Yang et al., 2018) show effective use of recurrent neural networks to exploit semantic representations by treating documents as a sequence of words or sentences for classification. In this work, we explore several neural baseline models and use both LSTM (Hochreiter and Schmidhuber, 1997) and KimCNN architectures for knowledge distillation experiments.

Non-neural models. Logistic regression (LR) and support vector machines (SVM) trained on tf–idf vectors form efficient and effective baselines for document classification. Adhikari et al. (2019) show LR and SVM surpass most of the neural baselines on multiple datasets, questioning the need for employing neural networks to model syntactic structure for document classification. Here, we explore both LR and SVMs, and we perform knowledge distillation experiments using an LR model.

Large-scale pre-training. Recent work (Howard and Ruder, 2018; Devlin et al., 2019; Yang et al., 2019) has demonstrated the effectiveness of large-scale pre-training for NLP tasks. In this work, we use BERT as a representative of this approach and demonstrate the power of fine-tuned BERT on document classification (termed DocBERT).

2.2 Knowledge Distillation

Knowledge distillation (KD; Hinton et al., 2015; Ba and Caruana, 2014) is an effective model-agnostic approach to model compression, where an efficient *student* model captures the knowledge learnt by privileged but cumbersome *teacher* model(s). The knowledge transfer takes place by forcing the *student* to mimic the soft target probabilities of the teacher. Hinton et al. (2015) highlight that it is in the interest of the generalizability of the *student* model to capture the exact class probabilities from a better model, the *teacher*. In supervised settings, the *student* is trained using a distillation objective in combination with the classification objective:

Dataset	C	N	W	S
Reuters	90	10,789	144.3	6.6
AAPD	54	55,840	167.3	1.0
IMDB	10	135,669	393.8	14.4
Yelp 2014	5	1,125,386	148.8	9.1

Table 1: Summary of the datasets. C denotes the number of classes in the dataset, N the number of samples, and W and S the average number of words and sentences per document, respectively.

$$\mathcal{L} = \mathcal{L}_{classification} + \lambda \cdot \mathcal{L}_{distill} \tag{1}$$

where λ is a hyperparameter chosen to weigh the two different optimization objectives. The $\mathcal{L}_{classification}$ term is the task-specific classification loss, which is most often the cross-entropy loss between the logits of the *student* model and the target labels, while the distillation term $\mathcal{L}_{distill}$ quantifies the difference between the student predictions and the teacher. In this work, we use a fine-tuned BERT model as the teacher and experiment with various baseline architectures for the students. Following Hinton et al. (2015), we set $\mathcal{L}_{distill}$ to be equal to the Kullback–Leibler divergence between the class probabilities output by the *student* and the *teacher* BERT model.

3 Datasets

We use the following four datasets to evaluate BERT: Reuters-21578 (Reuters; Apté et al., 1994), arXiv Academic Paper dataset (AAPD; Yang et al., 2018), IMDB reviews, and Yelp 2014 reviews. Reuters and AAPD are multi-label datasets while documents in IMDB and Yelp '14 contain only a single label per document. Table 1 summarizes the statistics of these datasets.

For Reuters, we use the standard ModApté splits (Apté et al., 1994); for AAPD, we use the splits provided by Yang et al. (2018); for IMDB and Yelp, following Yang et al. (2016), we randomly sample 80% of the data for training and 10% each for validation and test.

4 Training and Hyperparameters

As a simple and straightforward adaptation of BERT models (Devlin et al., 2019) for document classification, we introduce a fully-connected layer over the final hidden state corresponding to the [CLS] input token. During fine-tuning, we optimize the entire model end-to-end, with the additional softmax classifier parameters $W \in \mathbf{R}^{K \times H}$, where H is the dimension of the hidden state vectors and K is the number of classes. We minimize the cross-entropy and binary cross-entropy loss for single-label and multi-label tasks, respectively. While fine-tuning BERT, we optimize the number of epochs, batch size, learning rate, and maximum sequence length (MSL; i.e., the number of tokens that documents are truncated to).

For knowledge distillation, we train the LSTM, KimCNN, and LR to capture the learnt representations from BERT$_{large}$ using the objective of the type shown in Equation (1). Depending upon the dataset, we use cross-entropy or binary cross-entropy loss as $\mathcal{L}_{classification}$, Equation (1). For $\mathcal{L}_{distill}$, following Hinton et al. (2015), we minimize the Kullback–Leibler (KL) divergence $\mathrm{KL}(p||q)$ where p and q are the class probabilities produced by the student and the teacher models, respectively.

To build an effective transfer set for distillation as suggested by Ba and Caruana (2014), we augment the training splits of the datasets by applying POS-guided word swapping and random masking, as in Tang et al. (2019), along with randomizing the order of the sentences of documents in the training set. The transfer set sizes for Reuters, IMDB and AAPD are 3×, 4×, and 4× their training splits, respectively; for Yelp2014, no data augmentation was performed due to computational restrictions. Refer to the appendix for further details regarding the training hyperparameters.

5 Results and Discussion

In Table 2, which shows our main results, we report the mean F$_1$ scores for multi-label datasets and accuracy for single-label datasets, along with the corresponding standard deviation across five runs. Due to their higher computational costs, we report the scores from only a single run per task for BERT$_{base}$ and BERT$_{large}$.

Rows 1–7 report the model quality of pre-BERT models (that do not take advantage of pre-training). As observed by Adhikari et al. (2019), LR and SVM trained with tf–idf vectors form effective baselines as they challenge many neural network-based baselines (e.g., HAN) on multiple datasets. This raises the question whether neural networks are a suitable architectural choice for document classification. However, at a much higher computational cost, the regularized LSTM (Adhikari et al., 2019) (row 7) achieves the best numbers for the class of models that do not exploit pre-training.

Consistent with Devlin et al. (2019), the BERT-based models achieve state-of-the-art results on all four datasets (see Table 2, rows 8 and 9), with the BERT$_{large}$ model consistently achieving the highest model quality (compared to BERT$_{base}$).

Surprisingly, distilled LSTM (KD-LSTM, row 10) achieves parity with BERT$_{base}$ on average for Reuters, AAPD, and IMDB. In fact, it outperforms BERT$_{base}$ (on both dev and test) in at least one of the five runs. For Yelp, we see that KD-LSTM reduces the difference between BERT$_{base}$ and LSTM, but not to the same extent as in the other datasets.

Next, we explore the limits of KD by further distilling BERT$_{base}$ all the way down to KimCNN (Kim, 2014) (a single-layer CNN) and LR. It is not surprising that these models don't come close to BERT$_{base}$ owing to their limited expressivity. However, interestingly, we see massive leaps in model quality of these models after distillation (rows 1–3; 11–12). Specifically for multi-label datasets, both these models beat or come close to HAN and SGM, which are far more complex models. To put things in perspective, LR is a simple fully-connected layer and KimCNN contains merely $\sim 0.4\%$ parameters of BERT$_{base}$. These results demonstrate that KD can yield a broad spectrum of baselines for varying computational costs, all of which can be useful depending on the requirements.

Table 3 emphasizes the scale of compression achieved during inference with the help of KD, yielding over $4000\times$ faster LR to $40\times$ faster but effective LSTM compared to BERT$_{base}$. We calculate the number of parameters (# params) of models and floating-point operations (# FLOPS) during inference on average for Reuters. Additionally, Figure 1 shows the comparison between the number of parameters and prediction quality on the vali-

#	Model	Reuters		AAPD		IMDB		Yelp '14	
		Val. F_1	Test F_1	Val. F_1	Test F_1	Val. Acc.	Test Acc.	Val. Acc.	Test Acc.
1	LR	77.0	74.8	67.1	64.9	43.1	43.4	61.1	60.9
2	SVM	89.1	86.1	71.1	69.1	42.5	42.4	59.7	59.6
3	KimCNN	83.5 ±0.4	80.8 ±0.3	54.5 ±1.4	51.4 ±1.3	42.9 ±0.3	42.7 ±0.4	66.5 ±0.1	66.1 ±0.6
4	XML-CNN	88.8 ±0.5	86.2 ±0.3	70.2 ±0.7	68.7 ±0.4	–	–	–	–
5	HAN	87.6 ±0.5	85.2 ±0.6	70.2 ±0.2	68.0 ±0.6	51.8 ±0.3	51.2 ±0.3	68.2 ±0.1	67.9 ±0.1
6	SGM	82.5 ±0.4	78.8 ±0.9	–	$71.0^{\dagger}$	–	–	–	–
7	LSTM	89.1 ±0.8	87.0 ±0.5	73.1 ±0.4	70.5 ±0.5	53.4 ±0.2	52.8 ±0.3	69.0 ±0.1	68.7 ±0.1
8	BERT$_{base}$	90.5	89.0	75.3	73.4	54.4	54.2	72.1	72.0
9	BERT$_{large}$	**92.3**	**90.7**	**76.6**	**75.2**	**56.0**	**55.6**	**72.6**	**72.5**
10	KD-LSTM	91.0 ±0.2	88.9 ±0.2	75.4 ±0.2	72.9 ±0.3	54.5 ±0.1	53.7 ±0.3	69.7 ±0.1	69.4 ±0.1
11	KD-KimCNN	90.0 ±0.3	87.0 ±0.2	72.7 ±0.4	70.6 ±0.1	49.0 ±0.2	48.3 ±0.3	66.5 ±0.1	66.2 ±0.0
12	KD-LR	87.0	83.7	73.1	71.3	43.8	43.3	62.7	62.3

Table 2: Results for each model on the validation and test sets. Best values are bolded. Rows 1–7 have been taken from Adhikari et al. (2019). Model names of type "KD-X" (rows 10-12) refer to X trained using knowledge distillation from the fine-tuned BERT$_{large}$ (row 9).

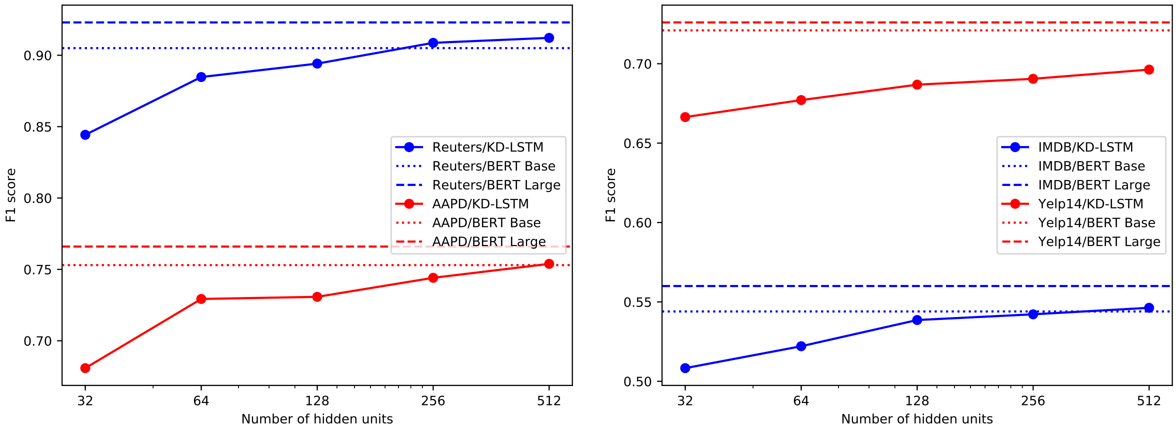

Figure 1: Effectiveness of KD-LSTM vs. BERT$_{base}$ and BERT$_{large}$

Model	# Params	# FLOPS
LR	3.3 (3%)	6.5 (4620×)
KimCNN	0.4 (0.4%)	26.0 (1150×)
LSTM	3.3 (3%)	780.9 (40×)
BERT$_{base}$	110	$\sim$30000

Table 3: # params for the models and # FLOPS for a single inference pass. Values are in millions. Figures in brackets are relative comparisons to BERT$_{base}$.

dation sets. These plots convey the effectiveness of the KD-LSTM model with different numbers of hidden units: 32, 64, 128, 256, and 512. We find that KD-LSTM, with just 256 hidden units (i.e., $\sim 1\%$ parameters of BERT$_{base}$) attains parity with BERT$_{base}$ on Reuters and IMDB, while for AAPD, 512 hidden units ($\sim 3\%$ of BERT$_{base}$) are enough.

6 Conclusion and Future Work

In this paper we improve baselines for document classification by fine-tuning BERT (DocBERT). Using DocBERT, we show the effectiveness of KD over a range of efficient models—a single-layer LSTM model, a single layer CNN, and a logistic regression trained on tf–idf. This provides us with a spectrum of baselines for varying tradeoffs in classification accuracy and complexity. In fact, we show that the distilled LSTM model achieves BERT$_{base}$ parity on a majority of datasets, using only $\sim 3\%$ parameters of the latter.

While distillation is an effective way to reduce computational cost during inference, it doesn't aid in reducing resources needed for training. Thus, methods for reducing the computational resources required while training deserve attention in future.

7 Acknowledgements

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

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Rethinking complex neural network architectures for document classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4046–4051.

Chidanand Apté, Fred Damerau, and Sholom M. Weiss. 1994. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251.

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems 27*, pages 2654–2662.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *arxiv/1503.02531*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–570.

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

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650.

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*.

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

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*, pages 5753–5763.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

A Appendix

A.1 Training Hyperparameters

While fine-tuning BERT, we optimize the number of epochs, batch size, learning rate, and maximum sequence length (MSL), the number of tokens that documents are truncated to. We observe that model quality is quite sensitive to the number of epochs, and thus the number must be tailored for each dataset. We train on Reuters, AAPD, and IMDB for 30, 20, and 4 epochs, respectively. Due to resource constraints, we train on Yelp for only one epoch. As is the case with Devlin et al. (2019), we find that choosing a batch size of 16, learning rate of 2×10^{-5}, and MSL of 512 tokens yields optimal model quality on the validation sets for all the datasets.

For distillation, we train an LSTM to capture the learnt representations from BERT$_{large}$ using the objective shown in Equation (1). We use a batch size of 128 for the multi-label tasks and 64 for the single-label tasks. We find the learning rates and dropout rates used in Adhikari et al. (2019) to be optimal even for the distillation process.

To build an effective transfer set for distillation as suggested by Hinton et al. (2015), we augment the training splits of the datasets by applying POS-guided word swapping and random masking for data augmentation, similar to Tang et al. (2019). For the distillation objective given in Equation (1), we use a λ of 1 for multi-label datasets and 4 for single-label datasets.

A.2 Hyperparameter Analysis for DocBERT

MSL analysis. A decrease in the maximum sequence length (MSL) corresponds to only a minor loss in F$_1$ on Reuters (see top-left subplot in Figure 2), possibly due to Reuters having shorter documents. On IMDB (top-right subplot in Figure 2), lowering the MSL corresponds to a drastic fall in accuracy, suggesting that the entire document is necessary for this dataset.

On the one hand, these results appear obvious. Alternatively, one can argue that, since IMDB contains longer documents, truncating tokens may hurt less. The top two subplots in Figure 2 show that this is *not* the case, since truncating to even 256 tokens causes accuracy to fall lower than that of the much smaller LSTM$_{reg}$ (see Table 2). From these results, we conclude that any amount of truncation is detrimental in document classification, but the level of degradation may differ.

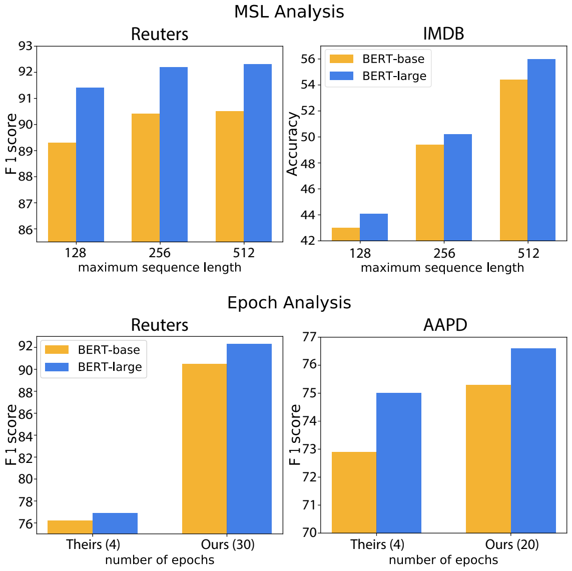

Figure 2: Results on the validation set from varying the MSL and the number of epochs.

Epoch analysis. The bottom two subplots in Figure 2 illustrate the F$_1$ score of BERT fine-tuned using different numbers of epochs for AAPD and Reuters. Contrary to Devlin et al. (2019), who achieve the state of the art on small datasets with only a few epochs of fine-tuning, we find that smaller datasets require many more epochs to converge. On both the datasets (see Figure 2), we see a significant drop in model quality when the BERT models are fine-tuned for only four epochs, as suggested in the original paper. On Reuters, using four epochs result in an F$_1$ worse than even logistic regression (Table 2, row 1).