Does local pruning offer task-specific models to learn effectively ?

Abhishek Kumar Mishra
Dept. of Electrical & Computer Engineering
Drexel University
Philadelphia, PA, USA
am4862@drexel.edu

Mohna Chakraborty
Dept. of Computer Science
Iowa State University
Ames, IA, USA
mohnac@iastate.edu

Abstract

The need to deploy large-scale pre-trained models on edge devices under limited computational resources has led to substantial research to compress these large models. However, less attention has been given to compress the task-specific models. In this work, we investigate the different methods of unstructured pruning on task-specific models for Aspect-based Sentiment Analysis (ABSA) tasks. Specifically, we analyze differences in the learning dynamics of pruned models by using the standard pruning techniques to achieve high-performing sparse networks. We develop a hypothesis to demonstrate the effectiveness of local pruning over global pruning considering a simple CNN model. Later, we utilize the hypothesis to demonstrate the efficacy of the pruned state-of-the-art model compared to the over-parameterized state-of-the-art model under two settings, the first considering the baselines for the same task used for generating the hypothesis, i.e., aspect extraction and the second considering a different task, i.e., sentiment analysis. We also provide discussion related to the generalization of the pruning hypothesis.

1 Introduction

Significant progress in deep neural networks (DNN) over the last decade (Liu et al., 2017) and access to unlimited online and offline data has revolutionized the research in Natural Language Processing (NLP). Neural-based language models (LMs) (Devlin et al., 2019; Brown et al., 2020) can utilize large volumes of data and discover patterns that can be used to facilitate various downstream tasks (Xu et al., 2019; Sun et al., 2019; Wu and He, 2019; Li et al., 2019; Dai et al., 2020). However, the development and deployment of such LMs require extensive resources that amplify the costs in industry settings and questions the easy deployment of such models onto low resource capable embedded devices such as mobile phones (Wu and He, 2019; Yang et al., 2017). For instance, pre-trained transformer-based LMs such as BERT (Devlin et al., 2019) have demonstrated state-of-the-art results for various applications such as machine reading comprehension, information retrieval, and question answering by extracting contextualized word embedding or fine-tuning BERT for specific functionality (Xu et al., 2019; Li et al., 2019; Dai et al., 2020). However, these models are over-parametrized and thus are memory hungry and time-intensive to deploy on resource-constrained devices. Therefore, it is crucial to develop energy-efficient and cost-effective models for use in production.

Applications in the real world are task-oriented, with a demand for resource-efficient models. So, the models are required to use fewer parameters. Given the need to build smaller task-specific architectures to save memory footprint and computational burden (Henderson et al., 2020; Bender et al., 2021), one popular solution is pruning, a well-vetted topic in computer vision. Pruning (Karnin, 1990) is a compression technique that systematically removes less significant parameters from an existing network to produce a smaller compressed model with similar performance comparing the larger model. The evolution of DNN has lead to the rise of research in pruning, even though the concept of pruning has existed for a long time. Abundant research has been conducted on compressing deep learning-based architectures for computer vision tasks (Li et al., 2016); however, few works have been proposed for compressing the task-specific models in NLP (Liu et al., 2018a).

In this work, we aim to sparsify the models for ABSA tasks. ABSA aims to capture the opinion of the reviewer towards specific aspect in a review. In product-based reviews (reviews for restaurants, websites, etc.), the aspect term describes the attribute of a product, and the opinion term captures the sentiment expressed towards the aspect term.

Proceedings of the Student Research Workshop associated with RANLP-2021, pages 118–125,
held online, Sep 1–3, 2021.
https://doi.org/10.26615/issn.2603-2821.2021_017

An example of the product-based review is depicted in Figure 1. In the review "The food is great however the service is poor." the aspect terms are *food* and *service* and the opinion terms are *great* and *poor*. The sentiment captured by the opinion terms is *positive* and *negative* respectively. Aspect extraction is considered as a sequence labeling task to consider the span of aspect terms. Each term in the span is assigned a label following BIO scheme (Ramshaw and Marcus, 1999) where B and I indicate the beginning and inside of the span, respectively, and O indicates outside of the span.

Review: *"The* ***food*** *is great however the* ***service*** *is poor."*

Target Terms: *Food, Service*
Sentiment: *Positive, Negative*

Figure 1: Example of a restaurant review

A complex model slows down the speed of inference, and thus a simple model is always preferred over these highly sophisticated architectures (Xu et al., 2018). To this end, we propose pruning a simple Convolutional Neural Network (CNN) (LeCun et al.; Wróbel et al., 2018), trained on general embedding and show that it produces promising results for ABSA tasks. In this paper, we perform an in-depth analysis of local and global pruning to sparsify architectures needed for real-world applications and empirically demonstrate that local pruning offers desirable sparsity with less compromising in performance, inferring that it is more practical in addressing real-world tasks.

Our contributions are as follows:

1. A meta-analysis of pruning a language model used for ABSA tasks.

2. Empirically demonstrates the effectiveness of local pruning over global pruning under the unstructured setting.

3. Empirically illustrates the possibility of generalization of pruning hypothesis and discuss our observations to pave the way for future research in the direction.

The rest of the paper is organized as follows: In Section 2, we provide an overview of related works in the field. In Section 3, we discuss the methodology proposed. Section 4 details the experimental setup, datasets, and results of the experiments. Section 5 concludes the paper and discusses the possible extensions of this work.

2 Related Works

2.1 *Aspect-based Sentiment Analysis*

Analysis of the sentiment expressed by a reviewer for a review has been studied in the past under different settings using supervised (Xu et al., 2019), semi-supervised (Dai and Song, 2019), and unsupervised (He et al., 2017) approaches. To exploit the full potential of supervised approaches, a large amount of labeled data is required, which is expensive to obtain. To alleviate this problem, semi-supervised and unsupervised approaches emphasize understanding features directly from raw corpus. This work focuses on extracting the aspect terms for a given review in a supervised fashion.

2.2 *Pruning*

Due to the over-parameterized nature of the deep neural networks (Mao et al., 2017; Frankle and Carbin, 2019) which lead to several problems like high computational costs, larger memory needs, etc. several compression methods like pruning (Han et al., 2015; Guo et al., 2016), quantization (Courbariaux et al., 2016; Shu and Nakayama, 2017) and knowledge distillation (Hinton et al., 2015) are proposed. Among them, pruning has been an efficient and effective method to reduce the number of parameters without loss of accuracy significantly. Moreover, it helps achieve higher compression rates (Han et al., 2015). Empirically, it has been shown that pruning performs better for sparse models compared to dense models (Lee et al., 2020). Furthermore, there have been many works on pruning for computer vision applications (Han et al., 2015; Li et al., 2017; Molchanov et al., 2017) but very few works exist for natural language processing applications (Joulin et al., 2016; Shu and Nakayama, 2017). Most of the works in NLP applications use quantization for language model compression (Joulin et al., 2016; Shu and Nakayama, 2017; Zadeh et al., 2020; Zafrir et al., 2019) and only a few apply pruning techniques for the purpose (Liu et al., 2018a).

3 Methodology

In this section, we introduce the methodology adopted for this work. We employ magnitude-based unstructured pruning where individual connections are detached based on the magnitude (L_1) of synaptic weights (Li et al., 2016) and then use it to implement both local and global pruning. In unstructured pruning, (Mao et al., 2017), individual connections between neurons or filters of adjacent filters are detached from the network, whereas in structured pruning (Liu et al., 2018b), the entire neurons or filters are detached from the network. In local pruning, (Han et al., 2015), a substantial percentage of connections are detached by contrasting each connection to the other connections in the layer, while in global pruning (Lee et al., 2018), all parameters are put together across all the different layers, and then a global percentage of them are taken to prune.

We have used unstructured pruning over structured pruning considering the in-feasibility of structured pruning due to ample search space for pruning rates per layer (Renda et al., 2020) and for causing large accuracy loss also (Li et al., 2016).

Algorithm 1 Training and pruning

```
$W \leftarrow randomlyInitialize()$
$M \leftarrow \{1\}^{|W|}$
for $i = 0 : E$ do
    $W' \leftarrow weightUpdate(f(X; W))$
    $M' \leftarrow unstructuredPrune(M, L1(W'))$
    $W' \leftarrow Set(f(X; M' \odot W'))$
    $W \leftarrow W'$
    $M \leftarrow M'$
end for
```

Algorithm 1 shows the proposed pruning algorithm where $f(X;W)$ refers to the neural network model, which is a collection of nested functions parameterized by weight W. W' is the updated weight, M denotes the mask, E represents the number of training epochs, and X is the training dataset. In the algorithm, we prune the language model using gradual pruning (Liu et al., 2018b; Renda et al., 2019). The result after training is the pruned model with desired sparsity. During the training phase, we first perform weight updates to obtain W', later $L1$-norm is applied on the updated weights to remove the least essential weights resulting in an updated binary mask M' that affixes a definite number of parameters to 0. The generated pruned model is $f(X; M' \odot W')$, and $M' \in \{0, 1\}^{|W'|}$ and $\odot$ is the element-wise product operator which helps to generate pruned weights. This enables the masked parameters to reactivate during training based on gradient updates. At last, we apply a gradual sparsification schedule with sorting-based weights to achieve desirable sparsification.

To show the effectiveness of the proposed hypothesis, we consider two CNN-based baselines with 4 and 6 convolutional layers respectively working on aspect extraction (AE) task. To study the possibility of pruning hypothesis generalization, we further consider two neural network-based architectures proposed by Xu *et al.* (Xu et al., 2018) and Li *et al.* (Li et al., 2019). The architectures proposed in these works use different language models thus enabling us to perform meta-analysis for our hypothesis. Xu *et al.* (Xu et al., 2018) uses a CNN-based model for aspect term extraction by employing double embeddings (domain + general) whereas Li *et al.* (Li et al., 2019) uses five different versions of BERT-based models to perform AE and sentiment analysis of the extracted aspect terms.

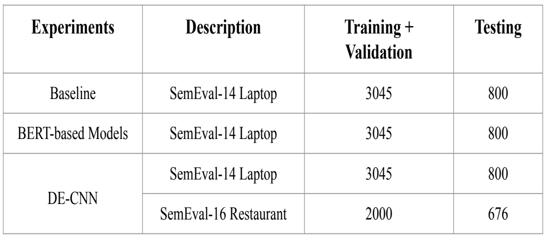
The code is implemented in Pytorch, and the code is available at the url[1].

4 Experiments

The effectiveness of the inferred hypothesis is tested considering two settings. In the first setting, we consider the baselines for the same task of aspect extraction, which is used to generate the hypothesis. In the second setting, we consider the baselines for a different sentiment analysis task to validate the generalization of the hypothesis. We use fastText (Bojanowski et al., 2017) for the general-purpose embedding.

4.1 *Dataset Overview*

Following the baseline papers (Xu et al., 2018; Li et al., 2019), we conduct our experiments on two benchmark datasets from SemEval challenges (Pontiki et al., 2014, 2016). The first dataset is from the laptop domain on subtask 1 of SemEval-2014 Task 4. The second dataset is from the restaurant domain on subtask 1 of SemEval-2016 Task 5. The statistics of the dataset are given in Figure 2.



Experiments	Description	Training + Validation	Testing
Baseline	SemEval-14 Laptop	3045	800
BERT-based Models	SemEval-14 Laptop	3045	800
DE-CNN	SemEval-14 Laptop	3045	800
	SemEval-16 Restaurant	2000	676

Figure 2: Dataset Overview

[1] https://github.com/abhishekkumarm98/Local_Vs_Global-Pruning.git

4.2 *Baselines*

To obtain the proposed hypothesis, we consider two different CNN-based models with 4 and 6 convolutional layers respectively. For both the models, after the convolutional layers, fully connected layer along with a softmax layer is applied. The architecture is shown in Figure 3. Furthermore, we consider general purpose embedding using fastText for both.

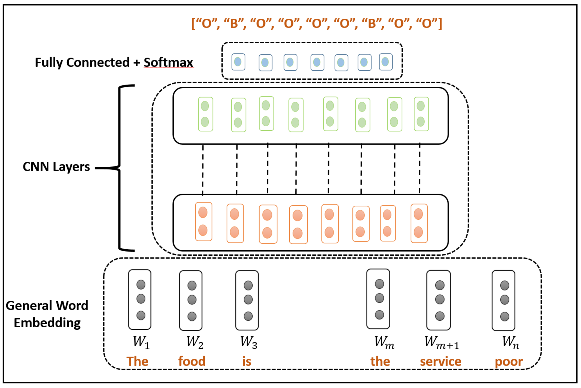


Figure 3: Baseline framework for (Conv-4 and Conv-6) layers where the input to the model is "The food is good however the service is poor".

The embedding layers in the LM provide vector representation at the character, word, or sentence level. Considering the distribution of parameters in the language models, a substantial percentage of parameters come from the embedding layer. During training, we apply different fractions of local and global pruning in an unstructured manner. The average f1-score of the resultant pruned model with respect to the fraction of weights pruned is shown in Figure 4 for SemEval-14 laptop dataset. For the experiments, the hyperparameter settings include epochs (200), batch size (128), learning rate ($1e$-4) with a learning rate scheduler dividing the learning rate by ten after each epoch.

From Figure 4, it can be observed that for both the models, applying local pruning resulted in a considerable average f1-score until 80% of the weights were pruned, whereas applying global pruning resulted in a substantial performance degradation after 40% of the weights were pruned. This observation validates our claim that local pruning is more efficient than global pruning for ABSA tasks.

4.3 *DE-CNN*

In order to validate the generalization of our proposed hypothesis, for the double-embedding CNN-based model, we have trained and pruned on both SemEval-14 laptop and SemEval-16 restaurant datasets by applying local and global pruning in an unstructured manner considering the same hyperparameter settings as proposed in the paper. We made a few modifications like changing the number of epochs to 300, introducing a learning rate scheduler, and early stopping to learn effectively and prevent overfitting. Results are shown in Figure 7.

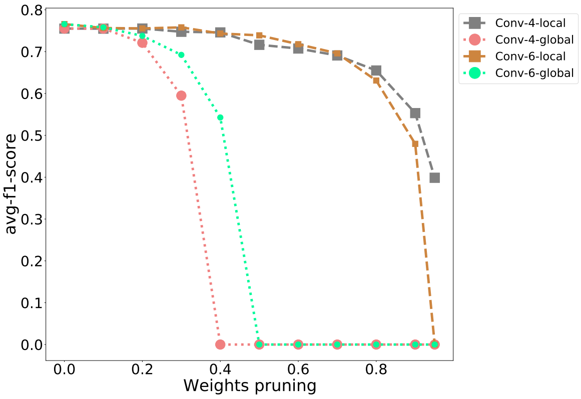


Figure 4: Performance of Conv-4 and Conv-6 models on applying local and global pruning on test set of SemEval-14 laptop.

From Figure 7, we can see that for both datasets, SemEval-14 laptop, and SemEval-16 restaurant, we have obtained a considerable f1-score on applying pruning locally until 80% of the weights were pruned, but the model's performance dropped drastically on applying pruning globally immediately after 30% of the weights were pruned.

In the literature, it has been shown that global pruning performs slightly better than local pruning (Blalock et al., 2020) which contradicts our observations. Our in-depth analysis shows that most of the works on pruning are performed for computer vision tasks, and their model architecture is different from the architecture of a LM. Standard model architectures like ResNet-50 (He et al., 2016), VGG-16 (Simonyan and Zisserman, 2014), extensively used in the computer vision domain, have a bunch of convolutional layers, batch normalization layers but lacks embedding layer, a crucial part of any LM. The general embedding layer of a language model contains substantial portions of the total parameters of the model. Figure 8 demonstrates the relative percentage of model parameters in each layer of the DE-CNN model.

We further analyzed the drastic drop in performance when we applied global pruning on the models considered in the baselines and DE-CNN. The results of our analysis are shown in Figure 5.

We have observed that sparsification occurs

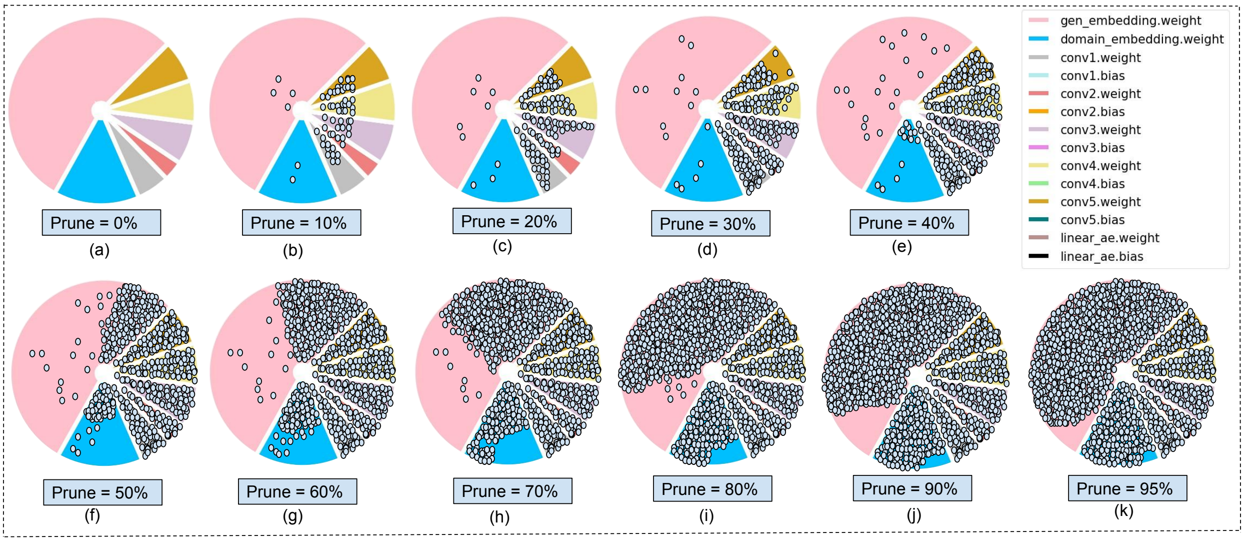


Figure 5: Spread of sparsification across each layer of DE-CNN with respect to different global pruning percentages.

more progressively for all layers except embedding layers as we increase the pruning percentage. From Figure 8, it can be observed that around 70% of the parameters belong to general and domain embedding layers. During global pruning, we pool all the parameters together and then take $L1$-norm to sort the absolute values in the decreasing order. The desired sparsification is applied to the sorted absolute values by setting the parameters to 0. From Figure 5 (b), it can be observed that sparsification has started in all layers, but as we progress, from Figure 5 (e), it can be observed that all layers except the embedding layers are sparsified completely. This leads to concluding that $L1$-norm causes the parameters of all the layers except the embedding layer to be smaller in magnitude resulting in the pruning of convolutional, bias, and fully connected layers ahead of embedding layers. This results in substantial decrement in models performance.

4.4 *BERT-based models*

In order to validate the generalization of our proposed hypothesis, we have trained and pruned the five different versions of BERT based models proposed in (Li et al., 2019) for an end to end ABSA task by applying local and global pruning in an unstructured manner considering the same hyperparameter settings as proposed in the paper. The result is demonstrated in Figure 9.

From Figure 9, it can be observed that applying local pruning resulted in a considerable performance for all the models until 60% of the weight pruning, whereas a substantial performance drop is observed after 30% of the weight pruning when global pruning is applied. The statistics of the crucial embedding layer of the model are shown in Figure 10. The reason for this performance drop is similar to what is observed for DE-CNN, where position type embedding and token type embedding layers are progressively sparsified compared to the word embedding layer as shown in Figure 6.

Furthermore, BERT has 200 layers where embedding, attention, and pooling layers have higher parameters than position and token type embedding layers, resulting in faster sparsification than other layers. Analyzing the phenomenon on the foundation of machine learning makes it analogous to garbage in, garbage out scenarios. Pruning most of the parameters by setting them to 0 results in an improper representation of the input, leading to performance deterioration observed for all the models. As per our analysis, the reason for better performance of local pruning compared to global pruning is because of giving more weightage during pruning to the layers with considerable parameters.

5 Conclusion and Future Works

This paper considers standard pruning techniques for compressing the model for two sub-tasks under ABSA tasks. We propose our hypothesis stating that local pruning is more effective than global pruning for aspect extraction. We empirically then demonstrate the validity of our hypothesis on two benchmark datasets for DE-CNN and show that pruned models can achieve comparable perfor-

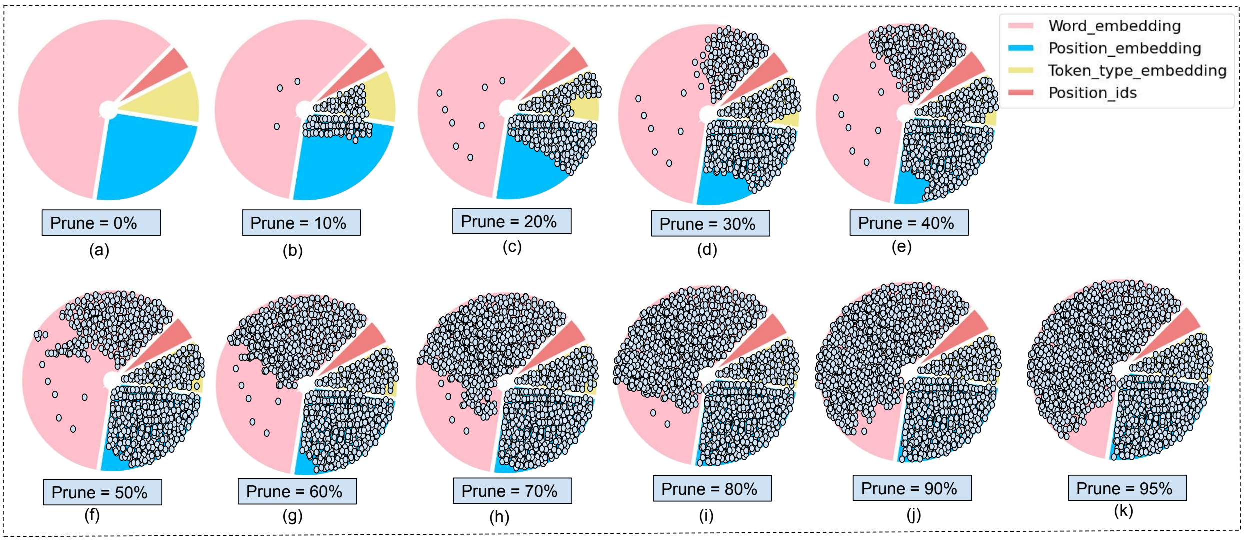


Figure 6: Spread of sparsification across each embedding layer of BERT with respect to different global pruning percentages.

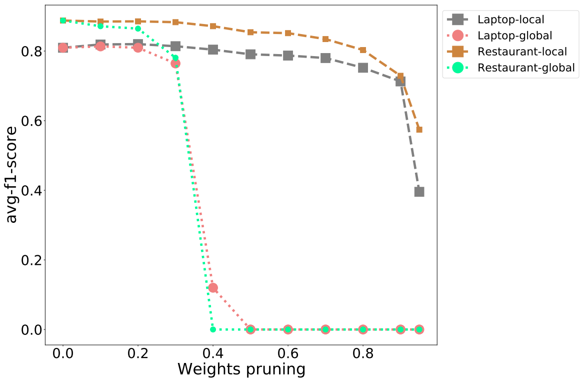


Figure 7: Performance of DE-CNN on test set of SemEval-14 laptop and SemEval-16 restaurant on applying local and global pruning.

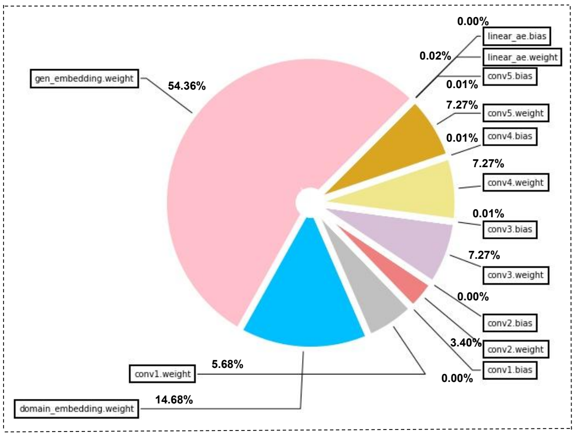


Figure 8: Relative percentage of parameters of each layer of DE-CNN.

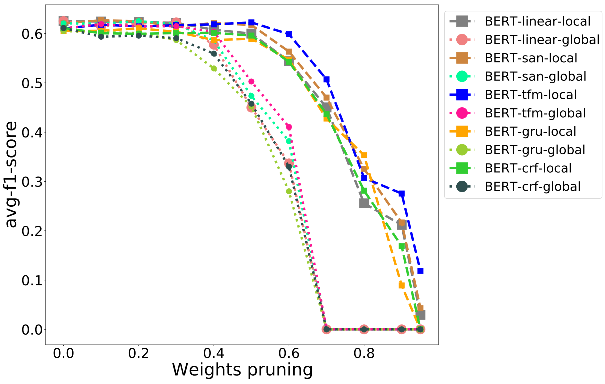


Figure 9: Performance of BERT (linear, crf, tfm, gru, san) on test set of SemEval-14 laptop on applying local and global pruning.

Embedding layer's name	Parameters	Percentage
Position_ids	511	0.0005%
Word_embedding	23,440,896	21.4012%
Position_embedding	393,216	0.3591%
Token_type_embedding	1,536	0.0014%

Figure 10: Details of BERT's crucial embedding layers.

mance comparing the original models. We further performed experiments on BERT-based models to verify the effectiveness of generalizing our hypothesis for the sentiment analysis task. In the future, we aim to explore different models on various tasks using other novel pruning-based techniques like global and local gradient magnitude.

Acknowledgement

We thank Adithya Kulkarni for his help and all the reviewers for their constructive comments.

References

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

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 preprint arXiv:1602.02830*.

Hongliang Dai and Yangqiu Song. 2019. Neural aspect and opinion term extraction with mined rules as weak supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5268–5277.

Zehui Dai, Cheng Peng, Huajie Chen, and Yadong Ding. 2020. A multi-task incremental learning framework with category name embedding for aspect-category sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6955–6965.

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.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks.

Yiwen Guo, Anbang Yao, and Yurong Chen. 2016. Dynamic network surgery for efficient dnns.

Song Han, Jeff Pool, John Tran, and William J Dally. 2015. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models.

Ehud D Karnin. 1990. A simple procedure for pruning back-propagation trained neural networks.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series.

Namhoon Lee, Thalaiyasingam Ajanthan, Stephen Gould, and Philip H. S. Torr. 2020. A signal propagation perspective for pruning neural networks at initialization.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*.

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

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2017. Pruning filters for efficient convnets.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting bert for end-to-end aspect-based sentiment analysis. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 34–41.

Liyuan Liu, Xiang Ren, Jingbo Shang, Xiaotao Gu, Jian Peng, and Jiawei Han. 2018a. Efficient contextualized representation: Language model pruning for sequence labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1215–1225.

Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. 2017. A survey of deep neural network architectures and their applications. *Neurocomputing*, 100(234):11–26.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. 2018b. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*.

Huizi Mao, Song Han, Jeff Pool, Wenshuo Li, Xingyu Liu, Yu Wang, and William J Dally. 2017. Exploring the regularity of sparse structure in convolutional neural networks. *arXiv preprint arXiv:1705.08922*.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning convolutional neural networks for resource efficient inference.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*, pages 19–30.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. "SemEval-2014 task 4: Aspect based sentiment analysis". In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Alex Renda, Jonathan Frankle, and Michael Carbin. 2019. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*.

Alex Renda, Jonathan Frankle, and Michael Carbin. 2020. Comparing rewinding and fine-tuning in neural network pruning. *arXiv preprint arXiv:2003.02389*.

Raphael Shu and Hideki Nakayama. 2017. Compressing word embeddings via deep compositional code learning.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. 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 380–385.

Krzysztof Wróbel, Marcin Pietroń, Maciej Wielgosz, Michał Karwatowski, and Kazimierz Wiatr. 2018. Convolutional neural network compression for natural language processing. *arXiv preprint arXiv:1805.10796*.

Shanchan Wu and Yifan He. 2019. Enriching pretrained language model with entity information for relation classification. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2361–2364.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 592–598.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. 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 2324–2335.

Tien-Ju Yang, Yu-Hsin Chen, and Vivienne Sze. 2017. Designing energy-efficient convolutional neural networks using energy-aware pruning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6071–6079. IEEE Computer Society.

Ali Hadi Zadeh, Isak Edo, Omar Mohamed Awad, and Andreas Moshovos. 2020. Gobo: Quantizing attention-based nlp models for low latency and energy efficient inference.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert.