Less Is More: Domain Adaptation with Lottery Ticket for Reading Comprehension

Haichao Zhu[†], **Zekun Wang**[†], **Heng Zhang**[†], **Ming Liu**[†‡], **Sendong Zhao**[†], **Bing Qin**[†‡*]
[†]Harbin Institute of Technology, Harbin, China
[‡]Peng Cheng Laboratory, Shenzhen, China
{hczhu,zkwang,hzhang,mliu,sdzhao,qinb}@ir.hit.edu.cn

Abstract

In this paper, we propose a simple few-shot domain adaptation paradigm for reading comprehension. We first identify the lottery subnetwork structure within the Transformer-based source domain model via gradual magnitude pruning. Then, we only fine-tune the lottery subnetwork, a small fraction of the whole parameters, on the annotated target domain data for adaptation. To obtain more adaptable subnetworks, we introduce self-attention attribution to weigh parameters, beyond simply pruning the smallest magnitude parameters, which can be seen as combining structured pruning and unstructured magnitude pruning softly. Experimental results show that our method outperforms the full model fine-tuning adaptation on four out of five domains when only a small amount of annotated data available for adaptation. Moreover, introducing self-attention attribution reserves more parameters for important attention heads in the lottery subnetwork and improves the target domain model performance. Our further analyses reveal that, besides exploiting fewer parameters, the choice of subnetworks is critical to the effectiveness. [1]

1 Introduction

Reading comprehension (Rajpurkar et al., 2016, 2018) obtains great attention from both research and industry for its practical value. State-of-the-art systems based on pre-trained language models (Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019; Dong et al., 2019; Joshi et al., 2020) have achieved remarkable performance on the task. Despite pretraining, they still rely on large amounts of annotated data (Rajpurkar et al., 2018; Trischler et al., 2017; Kwiatkowski et al., 2019) to reach the desired task performance. Manually collecting such high-quality datasets is costly and time-consuming, especially for cases that require specific domain knowledge. It hinders us from applying the data-driven solutions directly to scenarios or domains without sufficient annotation data. In this case, domain adaptation (Golub et al., 2017; Wang et al., 2019; Shakeri et al., 2020) is used to obtain a reasonable target domain performance.

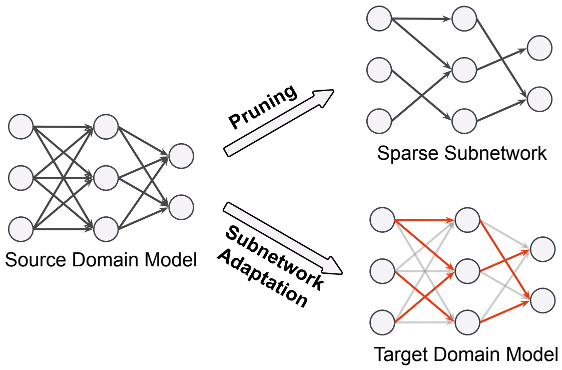


Figure 1: Domain adaptation with subnetworks of the source domain model. Various pruning methods can be used to find sparse subnetworks. Only the parameters (red arrow →) of the subnetworks are updated. The rest (grey arrow →) are frozen but used in inference.

Unsupervised domain adaptation (Wang et al., 2019; Cao et al., 2020) exploits the unlabeled context passages for adaptation. However, these methods have difficulties in adapting to the desiderata of questions and question-context reasonings in the target domain. In this paper, we focus on supervised domain adaptation for reading comprehension in the few-shot settings. We are devoted to transfer a model trained on a large amount of source domain data to the target domain with only limited annotated data. It is generally feasible to annotate a small amout of question answering pairs.

Typical reading comprehension models based on pre-trained language model contain at least hundreds of millions parameters, e.g., size of BERT-base is 110M. Previous works (Voita et al., 2019a;

[*] Corresponding author.

[1] The code is publicly available at https://github.com/haichao592/ALTER.

Michel et al., 2019; Sanh et al., 2020) show that dense neural networks are over-parameterized and considerable parameters of a trained model can be pruned with marginal or even no loss in performance. Meanwhile, "*The Lottery Ticket Hypothesis*" (Frankle and Carbin, 2019) argues that the initialization of over-parameterized neural networks contains sparse sub-network at initialization, which, when trained in isolation, rival the original network in task performance. On the other hand, our preliminary analysis (Figure 2) using an effective attribution method (Hao et al., 2020) shows that important attention heads are highly correlated across various domains.

In view of the over-parameterized source domain model and our preliminary findings on attention head dynamics, we assume fine-tuning a small fraction of deliberately selected parameters is both more efficient and more effective for few-shot domain adaptation. Specifically, we first prune the source domain model via magnitude pruning gradually. In addition, we introduce self-attention attribution (Hao et al., 2020) to reserve more parameters for important heads. The corresponding connections of the survived parameters after pruning depict the exact sparse structure of the lottery network. Then, we only fine-tune the lottery subnetwork, which consumes much less parameters, on the annotated target domain data for adaptation. The remaining parameters are frozen and will not be updated, but they also contribute to the predictions by participating in the forward computation.

Experimental results show that our method, exploiting small lottery subnetworks for few-shot domain adaptation, outperforms the full model fine-tuning on four out of five various domains with a range number of training examples. Further analyses reveal several intriguing findings. First, introducing attention head importance yields better lottery subnetworks in highly sparse regimes in the source domain. and improves the performance regardless of the sparsity. Secondly, the better source domain lottery subnetworks lead to the improved domain adaptation performance. Finally, in addition to using fewer parameters, the choice of subnetwork structure is critical to effectiveness.

2 Preliminary

2.1 The Transformer

Transformer (Vaswani et al., 2017) is a widely used model architecture that relies heavily on attention mechanism. A Transformer-based model consists of L stacked identical Transformer blocks. The model first embeds and then encodes the inputs through L-layer Transformer blocks $\mathbf{H}^l = \mathrm{Transformer}_l(\mathbf{H}^{l-1}), l \in [1, L]$. Each Transformer block consists of two sub-layers, a multi-head self-attention mechanism and a feed-forward network. A residual connection (He et al., 2016) followed by layer normalization (Ba et al., 2016) is employed around each of the two sub-layers.

The core component of a Transformer block is multi-head self-attention. For the l-th layer, the previous layer's output $\mathbf{H}^{l-1}$ is linearly projected to a triple of queries $\mathbf{Q}$, keys $\mathbf{K}$ and values $\mathbf{V}$ using parameter matrices $\mathbf{W}^l_Q, \mathbf{W}^l_K, \mathbf{W}^l_V \in \mathbb{R}^{d_k \times d_k}$ respectively. Then the attention of the i-th head is computed via:

$$\mathbf{A}_i = \mathrm{softmax}(\frac{\mathbf{Q}_i \mathbf{K}_i^\top}{\sqrt{d_k}}) \tag{1}$$

where d_k is the size of the hidden states. At last, the output of multi-head self-attention is $\mathrm{MultiHead}(\mathrm{H}^{l-1}) = [\mathbf{A}_1\mathbf{V}_1, \cdots, \mathbf{A}_h\mathbf{V}_h]\mathbf{W}^l_O$, where $\mathbf{W}^l_O \in \mathbb{R}^{d_k \times d_k}$, h is the number of heads, $[\cdot]$ means concatenation.

2.2 Self-Attention Head Importance

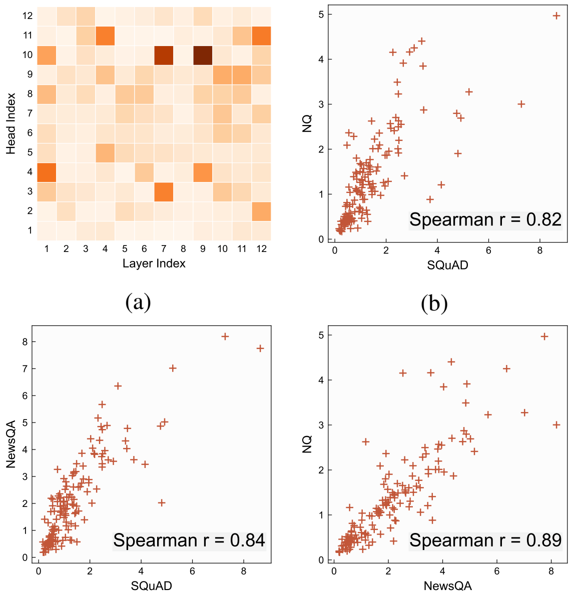
Many works (Clark et al., 2019; Kovaleva et al., 2019) have tried to interpret Transformer models' behaviors. Recently, Hao et al. (2020) propose a self-attention attribution (AttAttr) method by running an integrated gradients (Sundararajan et al., 2017) procedure over all the attention links. A higher attribution score indicates greater contribution to the model prediction.

Concretely, given input x of n tokens, the attribution score of each attention link within the i-th head is computed as:

$$\mathrm{Attr}(\mathbf{A}_i) = \mathbf{A}_i \odot \int_{\alpha=0}^{1} \frac{\partial \mathcal{F}(\mathrm{x}, \alpha\mathbf{A})}{\partial \mathbf{A}_i} \mathrm{d}\alpha \in \mathbb{R}^{n \times n}$$

where $\odot$ is element-wise multiplication, attention map $\mathbf{A}_i$ is computed as in Equation 1, $\mathbf{A} = [\mathbf{A}_1, \cdots, \mathbf{A}_h]$, and $\frac{\partial \mathcal{F}(\mathrm{x}, \alpha\mathbf{A})}{\partial \mathbf{A}_i}$ computes the gradient of model $\mathcal{F}(\cdot)$ along $\mathbf{A}_i$ with the manipulated attention weight matrix. Then, the importance score of the i-th attention head can be estimated via:

$$I_i = E_{\mathrm{x}}\left[\max(\mathrm{Attr}(\mathbf{A}_i))\right] \tag{2}$$



(a) (b) (c) (d)

Figure 2: (a) Estimated self-attention head importance on SQuAD v1.1. (b) - (d) Correlation of head importance scores between domain datasets. Each point represents the importance of the same attention head on two datasets. **Important heads are strongly correlated with high Spearman coefficient.**

2.3 The Lottery Ticket Hypothesis

The *Lottery Ticket Hypothesis* (Frankle and Carbin, 2019) suggests that we can find small and sparse subnetworks that rival the original network in performance, when trained in isolation from "lucky" initializations, often referred to as "winning lottery tickets". The connections of the winning lottery tickets are initialized to be particularly effective for training. Magnitude pruning (Han et al., 2015) is an effective method widely used to identify the winning lottery ticket by pruning the smallest magnitude weights.

2.4 Reading Comprehension Task and Domain Variance

In this work, we focus on extractive reading comprehension, which aims to extract a continuous span from the text context c as the answer a to a question q. It has been a prevalent format since SQuAD v1.1 (Rajpurkar et al., 2016) and widely adopted by several other reading comprehension datasets (Joshi et al., 2017; Trischler et al., 2017; Yang et al., 2018; Kwiatkowski et al., 2019) in various domains.

The differences between the domains are mainly derived from: a) the styles and the sources of the context passages, including Wikipedia, news articles, science articles, Web snippets, Tweets, b) the types of questions being asked, e.g., factoid, conversational, entity-centric, multi-hop reasoning, search queries, and c) the methodology under which the questions were collected, including manually written by crowdworkers, domain experts, and automatically mined from the web or search logs.

Our preliminary experiments explore the dynamics of important self-attention heads across different domains. We fine-tune BERT-base on each domain dataset independently to obtain domain-specific models. Then we employ ATTATTR, in Section 2.2, to get the importance scores of attention heads using Equation 2. We take three representative datasets, SQuAD v1.1 (Rajpurkar et al., 2016), NQ (Kwiatkowski et al., 2019) and NewsQA (Trischler et al., 2017), that differ in the sources of the context passages and question types. The heatmap of head importance on SQuAD v1.1 and the correlation of importance scores between each two of the three datasets are shown in Figure 2. Given the same BERT initialization, we can see that, despite the domain differences, the important heads are highly correlated. The preliminary results uncover the value of exploiting important heads for efficient domain adaptation.

3 Method

In this section, we describe our few-shot domain adaptation method for machine reading comprehension in detail. In the source domain, we have a model trained on a large-scale annotated dataset. We fine-tune BERT-base (Devlin et al., 2019), a representative Transformer-based pre-trained language model with tremendous number of parameters, as our source domain model. In the target domain, only limited annotated data, 1k examples at most, can be used for domain adaptation. The mismatch between a small amount of data and a large number of parameters makes it challenging to adapt all source domain model parameters to the target domain. Thus, we exploit a small fraction of deliberately selected parameters for domain adaptation by first identifying and then fine-tuning the lottery subnetwork.

3.1 Identifying the Lottery Network

Neural networks are over-parameterized (Allen-Zhu et al., 2019), a great fraction of the parameters are redundant and can be pruned with minimal or

Algorithm 1 Identifying the Lottery Subnetwork with Self-Attention Head Importance

```
Require:
    Source domain model $\mathcal{F}(x; \mathbf{M} \odot \theta_0)$
    Initial pruning mask $\mathbf{M} = 1^{|\theta_0|}$
    Target sparsity $s$, pruning frequency $\nabla t$ and steps $N$
    Importance factor $\lambda$
for $n \leftarrow 1$ to $N$ do
    Estimate attention head importance $I_n$                  ▷ Eq. 2
    $\hat{I}_n \leftarrow \lambda + (1-\lambda)\frac{I_n - \min(I_n)}{\max(I_n) - \min(I_n)}$      ▷ normalize
    Trim magnitudes with normalized importance score,
        $\hat{\theta}_{(n-1)\nabla t} \leftarrow$ AttrMagnitude($\theta_{(n-1)\nabla t}, \hat{I}_n$)
    $s_n \leftarrow s - s(1 - \frac{n}{N})^2$                 ▷ sparsity of step $n$
    Prune the lowest magnitudes parameters in group from
        $\hat{\theta}_{(n-1)\nabla t}$ to sparsity $s_n$
    Update the pruning mask $\mathbf{M}$
    Train the model for $\nabla t$ steps, producing $\mathcal{F}(x; \mathbf{M} \odot$
        $\theta_{n\nabla t})$
end for
Train the model util stopping criterion is met, producing
    $\mathcal{F}(x; \mathbf{M} \odot \theta_T)$
return Lottery Subnetwork $\mathbf{M}$
```

even no compromise in task performance. And *The Lottery Ticket Hypothesis* (Frankle and Carbin, 2019) suggests the existence of sparse subnetworks, trained from "lucky" initializations, that match the performance of the full model.

Magnitude pruning It is a simple and effective unstructured pruning method that prunes the smallest magnitude parameters (Han et al., 2015), which also used to find the winning lottery ticket (Frankle and Carbin, 2019). It requires several tricks to find lottery tickets for complicated architectures (Morcos et al., 2019). In our work, we employ a simple gradual pruning algorithm without iteratively rewinding parameters. It prunes a portion of the parameters each time and gradually increases the sparsity of the model. Training between pruning steps allows the model to recover from the pruning-induced task performance degradation. We follow Zhu and Gupta (2018) but use a square sparsity scheduling for magnitude pruning. The corresponding connections of the survived parameters after pruning depict the exact sparse structure of the lottery network. For the Transformer-based source domain model, we only prune the parameter matrix of the linear projections and feed-forward networks, i.e., $\mathbf{W}^l_Q, \mathbf{W}^l_K, \mathbf{W}^l_V, \mathbf{W}^l_O, \mathbf{W}^l_{FI}, \mathbf{W}^l_{IF}$, and keep the rest intact.

Pruning Strategy Pruning can be performed in two different ways: locally and globally. In local pruning, parameters magnitudes are compared within each parameter matrix separately, such that every parameter matrix will have the same fraction of pruned parameters. In global pruning, all parameters are pooled together prior to pruning, allowing the pruning fraction to vary across parameter matrices and layers.

Considering the intrinsic metric for magnitude pruning, the component importance may be overwhelmed by parameter magnitudes in global pruning. In cases that more parameters are pruned in important components due to their relative lower magnitudes. We observe that the magnitudes of Transformer parameter matrices are distributed uniformly across layers, but distantly across parameters matrices. Therefore, we propose a "divide-and-conquer" group pruning strategy, which divide the parameter matrices in groups according to their mean magnitudes and prune locally inter-group and globally intra-group.

Pruning with Self-Attention Head Importance Sanh et al. (2020) points that magnitude pruning is effective, but it is insufficient to determine the parameter importance using magnitude alone. Meanwhile, in Section 2.4, we find that attention heads are not equally important to the model predictions, and the important heads are highly correlated across various domains.

Thus, we introduce self-attention attribution (AttAttr; Hao et al., 2020) into magnitude pruning to identify more adaptable subnetworks when the sizes remain identical. In each pruning step, we first estimate the importance scores I of all attention heads using Equation 2. Then we scale the importance scores with $\mathrm{MinMax}(\lambda, I)$ normalization, where λ is the importance factor that negatively indicates the intensity of importance intervention. At last, we scale the parameters magnitudes accordingly, which may reverse the rankings previously determined by the magnitudes alone. Note that the parameters of an attention head are scattered in four parameter matrices. We apply the same importance scores to each parameter matrix and the slices of the same head are scaled identically within a layer.

In conclusion, we reserve more parameters for important heads, which are highly correlated across domains, due to its high self-attention attribution scores under the same pruning budget, and vice versa. That is we have lottery networks that are potentially more adaptable to target domains. Our lottery networks identification method is shown in Algorithm 1.

3.2 Adapting the Lottery Subnetwork

In Section 3.1, we have identified the sparse structure of the lottery subnetwork for adaptation. When adapting to the target domain, we use the original source domain model parameters and only update the lottery subnetwork parameters with limited annotated data, 1k examples at most. In this way, we adapt from an integrated source domain model without potential performance loss induced by pruning. Note that the pruned parameters are frozen and will not be updated, but they participate in the forward computation.

4 Experimental Setup

4.1 Datasets

We simulate few-shot domain-adaptation scenarios by sampling subsets from larger training sets. We use SQuAD v1.1 (Rajpurkar et al., 2016) as the resource-rich source domain and five various datasets, in Table 1, as the target domains:

SQuAD v1.1 (Rajpurkar et al., 2016): Crowdworkers are shown with Wikipedia paragraphs and ask questions with extractive answers. We use the default splits of training and development sets, containing $87,599$ and $10,570$ examples respectively.

NewsQA (Trischler et al., 2017): NewsQA is crowdsourced based on CNN news articles. Questions are asked by only seeing the article's headline and summary instead of the full article. We use the MRQA Shared Task (Fisch et al., 2019) version.

TriviaQA (Joshi et al., 2017): Question and answer pairs are sourced from trivia and quiz-league websites. We employ MRQA Shared Task version where the contexts are web snippets and documents from the Bing search engine.

TweetQA (Xiong et al., 2019): TweetQA is crowdsourced by gathering tweets used by journalists to write news articles as the context. We only keep the extractive questions and obtain $7,108$ training examples and 883 development examples.

NaturalQuestions (Kwiatkowski et al., 2019): Questions are users' information-seeking queries from the Google search engine logs. Answers are annotated in a retrieved Wikipedia page by crowdworkers. We use the MRQA Shared Task version of NQ, only containing examples that have short answers, and use the long answer as the context.

Dataset	Context	Question	Q⊥C	Train	Dev
SQuAD	Wikipedia	Crowd	✗	87,599	10,507
NewsQA	News articles	Crowd	✓	74,160	4,212
TriviaQA	Web snippets	Trivia	✓	61,688	7,785
TweetQA	Tweets	Crowd	✗	7,108	883
NQ	Wikipedia	Queries	✓	104,071	12,836
QuAC	Wikipedia	Crowd	✓	51,695	4,368

Table 1: Characteristics and splits of different datasets. ✓ in Q⊥C indicates that the question is collected independently from the context passage.

QuAC (Choi et al., 2018): QuAC contains conversational questions in the context of multi-turn information-seeking dialogues. We filter out yes/no questions and unanswerable questions.

4.2 Baselines

We compare our method, ALTER (**A**daptable **Lotter**y), against the following baselines:

Zero-Shot We apply the source domain model to the target domain without adaptation.

Fine-tuning We fine-tune the full source domain model on the target domain data.

EWC Elastic Weight Consolidation (Kirkpatrick et al., 2017) is a regularization algorithm that constrains parameters to stay close to their original values and prevents large deviations.

Layer Freeze We only fine-tune the top layers of the source domain model on the target domain data and freeze the rest.

Adapter Houlsby et al. (2019) proposes adapters for efficient transferring by adding only a few trainable parameters. We add adapters within transformer blocks and only update adapters.

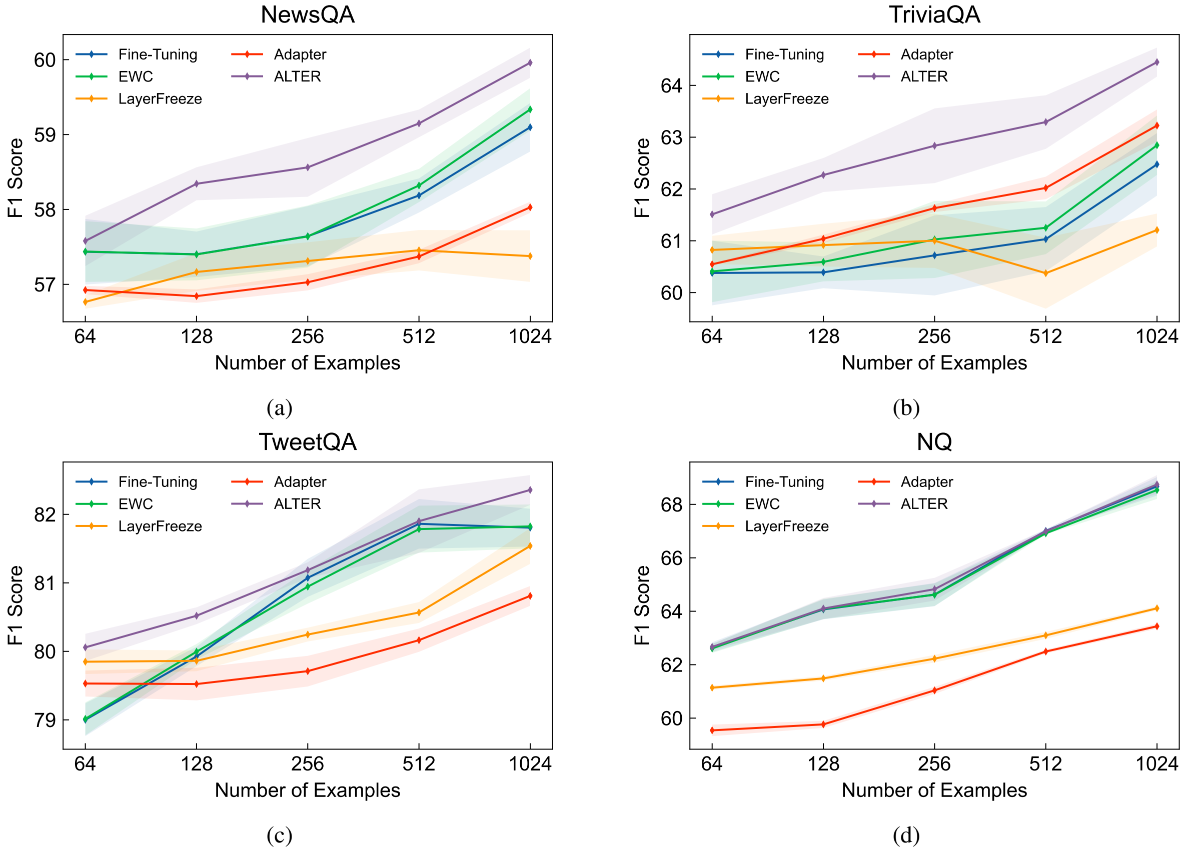
4.3 Implementation Details

We experiment with BERT-base-uncased [2] (Devlin et al., 2019), a Transformer-based pre-trained model with roughly 110M parameters. Fine-tuning embedding layer in the target domain yields no consistent differences. We thus freeze the embedding layer and reported sparsity percentages are relative to model without embedding layer, i.e., 84M parameters. We set maximum sequence length 384 with document stride 128. Adam (Kingma and Ba, 2015) with linear learning rate decay is used for optimization. The source domain model is BERT

[2] We use PyTorch (Paszke et al., 2019) implementation from Hugging Face Transformer library (Wolf et al., 2020).

Model	Training Parameters	NewsQA	TriviaQA	TweetQA	NQ	QuAC
		EM/F1	EM/F1	EM/F1	EM/F1	EM/F1
Zero-Shot	None	40.05/56.76	50.52/60.11	67.46/79.48	46.10/59.99	15.82 /37.31
Fine-tuning	84M	43.24/59.10	55.60/62.48	70.59/81.81	**55.23/68.68**	26.73/49.25
EWC	84M	43.44/59.34	55.95/62.85	70.48/81.82	55.09/68.54	26.82/49.37
LayerFreeze	21M	40.68/57.38	53.83/61.21	70.32/81.54	50.41/64.11	25.39/47.56
Adapter	20M	41.14/58.03	55.71/63.22	69.50/80.81	49.45/63.44	24.06/46.22
Alter	21M	**43.73/59.78**	**57.47/64.45**	**71.18/82.31**	54.62/68.17	**27.50/49.50**
Full Data	84M	52.18/66.95	64.44/70.26	68.59/80.58	67.03/78.89	38.37/60.38

Table 2: EM and F1 score across all domains when the number of training examples is 1024. Fine-tuning and EWC updates the full model. LayerFreeze, Adapter and Alter have the roughly the same capacity. Zero-shot applies the source domain model without adaptation and provides a lower bound. Full Data is obtained using the full training set without adaptation. The highest scores in each domain are marked in **bold**.



(a)

(b)

(c)

(d)

Figure 3: F1 score of Alter and four baselines on 4 datasets with various numbers of target domain examples. Error bars represent mean ± standard deviation across five trials. **Alter performs better than other methods or competitively with fewer parameters.**

fine-tuned on SQuAD v1.1 with learning rate of 3e-5 and batch size 12 for 2 epochs. We search for the best learning rate out of $[3\text{e-}5, 6\text{e-}5]$ and select epoch out of $[2, 3]$ in the target domain. Attention head importance are estimated with 200 source domain examples, using model predictions instead of the gold answers. Importance factor λ is set to 0.2 for the best performance.

5 Results and Analyses

5.1 Domain Adaptation Results

Table 2 shows the exact match (EM) and F1 scores on five target domains with 1024 training examples. We use magnitude pruning together with self-attention head importance to identify the lottery subnetworks, which contain 21M parameters and

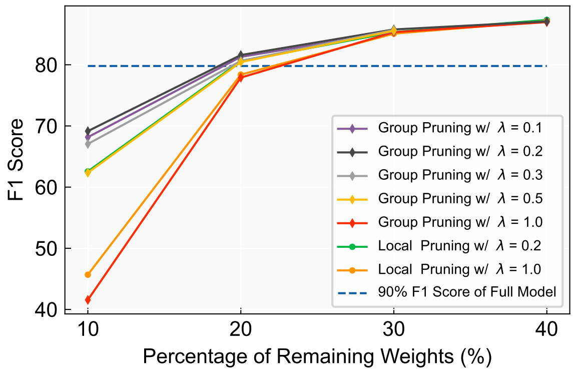


Figure 4: Pruning with and without self-attention head importance. Horizontal line indicates 90% F1 score of the full model. Sparsity percentage are relative to BERT-base with 84M parameters. λ is importance factor. For both local pruning and group pruning, **attention head importance scores help identify better lottery networks at high sparsity levels.**

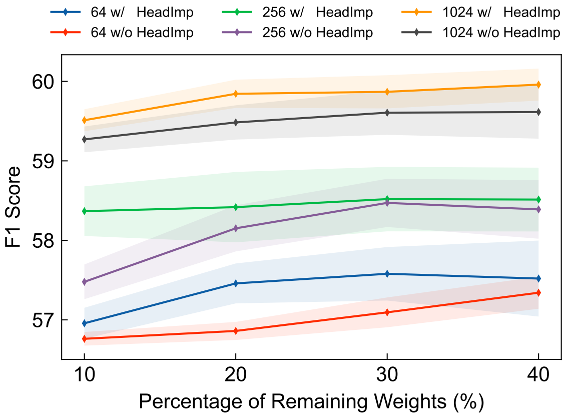


Figure 5: F1 score of subnetworks identified with and without self-attention head importance on NewsQA at different sparsity levels. Error bars represent mean ± standard deviation across five trials. **Subnetworks containing more parameters of important attention heads perform better in the target domain.**

correspond to approximately 25% of all parameters. We fine-tune the top 3 layers in LayerFreeze baseline and set the adapter size to 128. Experimental results show that ALTER outperforms the full model fine-tuning baseline and EWC regularized baseline on four out of five target domains. LayerFreeze and Adapter use roughly the same number of parameters as our method. However, they both perform worse than the fine-tuning baseline in most cases, which indicates that the structure to accommodate parameters is important. ALTER of this size performs worse than fine-tuning baseline on NQ, but competitively when using 42M parameters.

In Figure 3, we plot the F1 score of ALTER against all baselines on four domains in a range number of few-shot settings. EWC performs competitively with the fine-tuning baseline and occasionally yields slightly better results. Our method is orthogonal to EWC and can be exploited together, which we leave it to the future work. As in Table 2, FreezeLayer and Adapter are less competitive, except for TriviaQA in Figure 3b. However, Adapter consistently performs more robustly than other methods. We can clearly see that ALTER obtains superior performance in three domains with 64 to 1024 examples. Results on NQ are shown in Figure 3d, ALTER matches the fine-tuning baseline with only a half of the parameters. Besides, we present our method with the best performing lottery subnetworks and the optimal sizes in each domain are not identical. We find 20% $\sim$ 30% parameters are satisfactory, the only exception is 50% for NQ. In conclusion, ALTER is shown to be both effective and efficient for few-shot domain adaptation.

5.2 Analyses

Does structure-aware pruning deliver better lottery subnetworks? In Figure 4, the F1 scores of lottery networks identified with or without attention head importance in the source domain are shown. Since local pruning and global pruning perform competitively, we only present the results using local pruning and our group pruning (Section 3.1). At low sparsity (more than 30% of remaining weights), two pruning methods perform equally well and head importance has little effect in varying F1 score. However, at high sparsity, pruning with head importance maintains the performance of subnetworks within 90% of the full model with only 20% of remaining parameters. Meanwhile, group pruning works better with structure-aware importance determination.

Next, we investigate to what extent should we exploit attention head importance scores for pruning. Smaller importance factors λ in Algorithm 1 means that we can alter the parameters magnitudes more dramatically. That is the importance of parameters is more determined by its attention head importance. In Figure 4, we find that setting λ to 0.2 consistently leads to better lottery subnetworks of different sizes.

Do better lottery subnetworks improve domain adaptation performance? We have shown that attention head importance does help identify better lottery subnetworks in the source domain. Does the

Method	NewsQA EM/F1	TriviaQA EM/F1	TweetQA EM/F1
FINE-TUNING	40.59/57.40	53.13/60.39	68.23/79.93
RANDOM	40.98/57.72	54.45/61.98	68.57/80.24
MAGNITUDE	40.76/57.56	54.10/62.11	68.76/80.21
SALVAGE	40.86/57.67	54.39/61.75	68.82/80.24
ATTRHEAD	41.31/58.08	54.35/61.80	68.88/**80.39**
ALTER	**41.38/58.11**	**54.60/62.21**	**68.89**/80.35

Table 3: Performance of different subnetwork identification methods on three target domain datasets with 128 examples. The number of parameters are 21M, which corresponds to 25% of the full model size. **Structured attention head importance scores help identify better lottery subnetworks.**

better source domain performance lead to more efficient adaptation in the target domain? To answer this question, we present the difference of F1 score of lottery subnetworks identified with or without self-attention head importance in Figure 5. It shows consistent improvement with different number of target domain examples. The improvement tends to be magnified at higher sparsity, which is in tune with the trends in Figure 4.

What about other alternatives to lottery networks identification? We have investigated several heuristic methods to explore the choice of subnetwork structures for domain adaptation:

RANDOM chooses parameters to constitute subnetworks randomly.

MAGNITUDE selects the highest magnitudes parameters in one-shot.

SALVAGE reuses the pruned redundant parameters, which operates conversely with our method.

ATTRHEAD prunes the whole attention head with structured pruning, and applies unstructured magnitude pruning in feed-forward layers.

In Table 3, the sizes of subnetworks are identical. Methods in the second group work without structure importance priors. They perform similarly and outperform the full-model fine-tuning baseline surprisingly, which shows adapting all parameters to the target domain is not optimal when given few examples. We put the structure-aware methods in the third group. Comparing SALVAGE and ALTER, we find using important parameters instead of the redundant parameters are more effective. Results on ATTRHEAD show that high magnitude parameters in less important heads are also useful.

6 Related Work

Domain Adaptation and Generalization in MRC Previous domain adaptation works (Nishida et al., 2020) are mainly unsupervised and require plenty of unlabeled text. Most of them are devoted to generate synthetic questions (Golub et al., 2017). Adversarial training (Wang et al., 2019; Lee et al., 2019; Cao et al., 2020), self-training (Rennie et al., 2020) and several filtering methods (Shakeri et al., 2020; Rennie et al., 2020) are explored in this direction. But they have the inherent difficulty to accommodate the question and reasoning types desired in the target domain.

Several works have explored the domain generalization in reading comprehension. Talmor and Berant (2019), Khashabi et al. (2020) and Lourie et al. (2021) improve the generalization by training on multiple datasets. Su et al. (2020) introduces Adapters (Houlsby et al., 2019) to accommodate each domain. Theses method requires a quite amount of annotated data to work. We focus on more efficient few-shot domain adaptation. Ram et al. (2021) explores few-shot question answering via pre-training, which is orthogonal to our work.

Analyzing and Pruning Transformer Analyses (Clark et al., 2019; Mareček and Rosa, 2019; Voita et al., 2019b; Brunner et al., 2020; Hao et al., 2020) on Transformer mainly focus on understanding the multi-head self-attention mechanism. Michel et al. (2019); Voita et al. (2019a,b) show that most self-attention heads can be pruned with marginal performance loss. Structured pruning on more components are also explored (McCarley et al., 2019; Fan et al., 2020). We are inspired to treat self-attention heads unequally for domain adaptation. Unstructured magnitude pruning (Han et al., 2015) with tricks (Zhu and Gupta, 2018; Frankle et al., 2020) can reduce more parameters (Sanh et al., 2020; Gordon et al., 2020). In this work, we exploit both structured and unstructured pruning to find sparse structures.

Lottery Ticket in NLP The *Lottery Ticket Hypothesis* (Frankle and Carbin, 2019) is largely researched in Vision. Recent works (Yu et al., 2020; Prasanna et al., 2020; Chen et al., 2020) in NLP explore the existence of lottery subnetworks at pre-trained initialization and after training on downstream tasks. In our work, we identify and fine-tune lottery subnetworks for domain adaptation.

7 Conclusions

In this work, we propose ALTER, a simple and effective domain adaptation paradigm for few-shot reading comprehension. We exploit a small fraction of parameters of the over-parameterized source domain model to adapt to the target domain by first identifying and then fine-tuning the lottery subnetwork. We introduce self-attention attribution, an interpreting method for Transformer, to identify better subnetworks and improve the target domain performance. Further exploration on using several heuristic methods to reveal subnetwork structures find that subnetwork structures are critical to the effectiveness besides using fewer parameters.

Acknowledgement

We thank anonymous reviewers for their insightful feedback that helped improve the paper. This work is supported by the Science and Technology Innovation 2030 - "New Generation Artificial Intelligence" Major Project (2018AA0101901), the National Key Research and Development Project (2018YFB1005103), the Key Project of National Science Foundation of China (61632011), the National Science Foundation of China (61772156, 61976073), Shenzhen Foundational Research Funding (JCYJ20200109113441941), and the Foundation of Heilongjiang Province (F2018013).

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. 2019. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On identifiability in transformers. In *International Conference on Learning Representations*.

Yu Cao, Meng Fang, Baosheng Yu, and Joey Tianyi Zhou. 2020. Unsupervised domain adaptation on reading comprehension. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7480–7487. AAAI Press.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pre-trained BERT networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2174–2184. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering, MRQA@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 1–13. Association for Computational Linguistics.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. 2020. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3259–3269. PMLR.

David Golub, Po-Sen Huang, Xiaodong He, and Li Deng. 2017. Two-stage synthesis networks for transfer learning in machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 835–844, Copenhagen, Denmark. Association for Computational Linguistics.

Mitchell Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing BERT: Studying the effects of weight pruning on transfer learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 143–155, Online. Association for Computational Linguistics.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2020. Self-attention attribution: Interpreting information interactions inside transformer. *CoRR*, abs/2004.11207.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4364–4373. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Seanie Lee, Donggyu Kim, and Jangwon Park. 2019. Domain-agnostic question-answering with adversarial training. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 196–202, Hong Kong, China. Association for Computational Linguistics.

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. *CoRR*, abs/1907.11692.

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. UNICORN on RAINBOW: A universal commonsense reasoning model on a new multitask benchmark. *CoRR*, abs/2103.13009.

David Mareček and Rudolf Rosa. 2019. From balustrades to pierre vinken: Looking for syntax in transformer self-attentions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and*

Interpreting Neural Networks for NLP, pages 263–275, Florence, Italy. Association for Computational Linguistics.

JS McCarley, Rishav Chakravarti, and Avirup Sil. 2019. Structured pruning of a bert-based question answering model. *arXiv preprint arXiv:1910.06360*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ari Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. 2019. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Kosuke Nishida, Kyosuke Nishida, Itsumi Saito, Hisako Asano, and Junji Tomita. 2020. Unsupervised domain adaptation of language models for reading comprehension. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 5392–5399. European Language Resources Association.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT plays the lottery, all tickets are winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3208–3229. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-shot question answering by pretraining span selection. *CoRR*, abs/2101.00438.

Steven Rennie, Etienne Marcheret, Neil Mallinar, David Nahamoo, and Vaibhava Goel. 2020. Unsupervised adaptation of question answering systems via generative self-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1148–1157, Online. Association for Computational Linguistics.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online. Association for Computational Linguistics.

Lixin Su, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2020. Continual domain adaptation for machine reading comprehension. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 1395–1404. ACM.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4396–4406, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019b. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang. 2019. Adversarial domain adaptation for machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2510–2520. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. TWEETQA: A social media focused question answering dataset. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5020–5031. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S. Morcos. 2020. Playing the lottery with rewards and multiple languages: lottery tickets in RL and NLP. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Michael H. Zhu and Suyog Gupta. 2018. To prune, or not to prune: Exploring the efficacy of pruning for model compression. In *International Conference on Learning Representations, Workshop Track Proceedings*.