Adapting a Language Model While Preserving its General Knowledge

Zixuan Ke[1], Yijia Shao[2], Haowei Lin[2], Hu Xu[3], Lei Shu[1*] and Bing Liu[1]

[1]Department of Computer Science, University of Illinois at Chicago
[2]Wangxuan Institute of Computer Technology, Peking University
[3]Meta AI
[1]{zke4,liub}@uic.edu
[2]{shaoyj, linhaowei}@pku.edu.cn
[3]huxu@fb.com

Abstract

Domain-adaptive pre-training (or *DA-training* for short), also known as *post-training*, aims to train a pre-trained general-purpose language model (LM) using an unlabeled corpus of a particular domain to adapt the LM so that end-tasks in the domain can give improved performances. However, existing DA-training methods are in some sense *blind* as they do not explicitly identify what knowledge in the LM should be preserved and what should be changed by the domain corpus. This paper shows that the existing methods are sub-optimal and proposes a novel method to perform *a more informed adaptation* of the knowledge in the LM by (1) soft-masking the attention heads based on their importance to best preserve the general knowledge in the LM and (2) contrasting the representations of the general and the full (both general and domain knowledge) to learn an integrated representation with both general and domain-specific knowledge. Experimental results will demonstrate the effectiveness of the proposed approach.[1]

1 Introduction

Pre-trained general-purpose language models (LMs) like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and GPT-3 (Brown et al., 2020) have become a standard component in almost all NLP applications. Researchers have also found that *domain-adaptive pre-training* (or *DA-training* for short) using an *unlabeled* corpus in a specific domain to adapt an LM can further improve the end-task performance in the domain (Gururangan et al., 2020; Xu et al., 2019a,b; Sun et al., 2019; Alsentzer et al., 2019). Note that *domain-adaptive pre-training* is also called *post-training* (Xu et al., 2019a).

Existing DA-training methods simply apply the *same* pre-training objective, i.e., the *mask language model* (MLM) loss, to further train an LM using a domain corpus. These methods are sub-optimal because they do not explicitly identify what should be preserved and what should be updated in the LM by the domain corpus.

This paper argues that a good DA-training method has two needs. On the one hand, the general language knowledge learned in the LM should be preserved as much as possible because the target domain data is typically not large enough to be sufficient to learn the general knowledge well. For example, some words and their contexts may appear infrequently in a particular domain. The knowledge about them cannot be learned accurately based on the domain data alone. When these words and contexts appear in an end-task, the system will have difficulties. Thus, we need to rely on the knowledge about them in the LM. Since existing DA-training updates the LM with little guidance, such useful general knowledge may be corrupted. On the other hand, due to polysemy (same word with different meanings in different domains) and the fact that different domains also have their special word usages and contexts, the LM should be specialized or adapted to the target domain. A good DA-training should balance these two needs to adapt the LM to the target domain with minimal corruption to the good general knowledge in the LM.

This paper proposes a novel technique to enable a more *informed adaptation* to (1) preserve the *general* knowledge in the LM as much as possible, and (2) update the LM to incorporate the *domain-specific* knowledge of the target domain as needed. The focus of the existing DA-training research has been on (2). As we argued above, (1) is also important as focusing only on (2) may destroy some useful general knowledge and produce sub-optimal results for end-tasks. To achieve (1), the system should constrain the gradient update of each attention head[2] based on its importance to the *general*

---

*Now at Google Research leishu@google.com
[1]https://github.com/UIC-Liu-Lab/DGA

[2]We will see in Sec. 4 that constraining the neurons in

knowledge so that the general knowledge in LM can be preserved as much as possible. With (1), (2) will be able to change the part of the general knowledge that needs to be updated to adapt the LM to suit the target domain.[3]

In this paper, we propose a novel model called DGA (**D**A-training - **G**eneral knowledge preservation and LM **A**daptation) for the purpose. The key idea of the proposed method is to preserve the general language knowledge in the LM while adapting the LM to a specific domain. However, it is not obvious how this can be done, i.e., how to find those parameters that are important for the general knowledge and how to protect them. This paper proposes a novel proxy-based method to achieve the objectives. It works as follows. DGA first estimates the importance of each attention head in the LM via the newly proposed *proxy KL-divergence loss* (Sec. 3.1). This importance score reflects how important each attention head is to the general knowledge. Based on the importance scores, it performs two key functions: The first function uses the scores to *soft-mask* (rather than binary-mask or completely block) the gradient update to prevent important general knowledge in LM from being unnecessarily corrupted. This is related to *pruning of unimportant attention heads* (Michel et al., 2019). However, pruning is not directly applicable to DA-training as we will show in Sec. 2. The proposed soft-masking constrains only the backward gradient flow in training. It is not necessary to soft-mask the forward pass in either training or inference. This is important because using the knowledge in the full network encourages maximal integration of pre-trained general knowledge and the target domain-specific knowledge. The second function contrasts the representation for the general knowledge in the LM and the full (including both the general and the domain-specific) knowledge to learn an integrated representation (Sec. 3.2).[4]

In summary, this paper makes two key contributions.

(1). It proposes the idea of *informed adaptation* to integrate the specialized knowledge in the target

---

other layers is unnecessary.

[3]This is very different from *continual learning* (CL) (Chen and Liu, 2018) as CL needs to preserve the past knowledge to deal with *catastrophic forgetting* (McCloskey and Cohen, 1989). DA-training can and should change/adapt the general knowledge in the original LM to suit the target domain.

[4]Contrasting the general and only the domain-specific knowledge gives poorer results (see Sec. 4.3) as it causes the two types of knowledge to split rather than to integrate.

domain into the LM with minimal corruption to the useful general knowledge in the original LM.

(2). It proposes a new model DGA with two novel functions to enable better DA-training. DGA estimates the attention head importance to protect the important general knowledge in the LM and integrates the specialized knowledge in the target domain into the LM through contrasting the general and the full knowledge.

To the best of our knowledge, none of these has been reported in the literature before.

Extensive experiments have been conducted in 6 different domains and on 10 baselines to demonstrate the effectiveness of the proposed DGA.

2   Related Work

Domain-adaptive pre-training (DA-training). Researchers have applied DA-training to many domains, e.g., reviews (Xu et al., 2019a,b), biomedical text (Lee et al., 2020), news and papers (Gururangan et al., 2020), and social media (Chakrabarty et al., 2019). However, they all use the same mask language model (MLM) loss. We argue that it is sub-optimal and it is also important to preserve the general knowledge in the LM as much as possible and integrate it with the target domain knowledge.

Network pruning as importance computation. It is known that many parameters in a neural network are redundant and can be pruned (Li et al., 2021; Lai et al., 2021). This has also been shown for pre-trained Transformer (Chen et al., 2020a; Lin et al., 2020; Gao et al., 2021b; Michel et al., 2019; Voita et al., 2019). A popular pruning method is to discard the parameters with small absolute values (Han et al., 2015; Guo et al., 2016). Other methods prune the network at a higher level. In a Transformer-based model, these include pruning the attention head (Michel et al., 2019; Voita et al., 2019; McCarley et al., 2019) and pruning sub-layers in a standard Transformer layer (Fan et al., 2020; Sajjad et al., 2020). However, the above methods are not directly applicable to us as we need to compute the head importance for the LM using unlabeled domain data, while the above approaches are all for *supervised end-tasks*. We propose to use a *proxy KL-divergence* loss for our purpose. Note that it is possible to prune other sub-layers in the Transformer. However, as shown in Sec. 4.3, estimating the importance for other layers does not improve the performance.

Contrastive learning. Contrastive learning (Chen

et al., 2020b; He et al., 2020) can learn good representations by maximizing the similarity of positive pairs and minimizes that of negative pairs:

$$\mathcal{L}_{\text{contrast}} = -\log \frac{e^{(\text{sim}(q_i, q_i^+)/\tau)}}{\sum_{j=1}^{N} e^{(\text{sim}(q_i, q_j^+)/\tau)}}, \quad (1)$$

where N is the batch size, τ is a temperature parameter, $\text{sim}(\cdot)$ is a similarity metric, and q_i and q_i^+ are representations for positive pairs x_i and x_i^+ (typically, x_i^+ is an augmented sample of x_i, e.g., generated via cropping, deletion or synonym replacement (Gao et al., 2021a)). In the *unsupervised contrastive loss*, the negative samples are the other samples in the batch, indicated in the denominator.

We mainly use contrasive loss to contrast the representations of the important general knowledge in the original LM and the full knowledge (both the general and domain-specific knowledge) to achieve a good integration of the general knowledge and the domain specific knowledge.

3 Proposed DGA System

As discussed earlier, DGA goes beyond the MLM loss to perform two more functions: (1) preserving the important general knowledge in the LM by soft-masking the attention heads based on their importance. This helps avoid potential corruption of the general knowledge in the LM in DA-training (Sec. 3.1). However, the challenge is how to identify the general knowledge in the LM and how to protect it. We will propose a method to do that. (2) encouraging the model to learn integrated representations of the target domain and the general knowledge in the LM (Sec. 3.2). It is also not obvious how this can be done. We propose a contrastive learning based method to do it. Figure 1 gives an overview of DGA.

3.1 Preserving General Knowledge by Soft-Masking Attention Heads

Multi-head attention. Multi-head attention is arguably the most important component in the Transformer model (Vaswani et al., 2017). We omit details of other parts and refer the reader to the original paper. Formally, let $\boldsymbol{x} = x^{(1)}, ..., x^{(T)}$ be a sequence of T real vectors where $x^{(t)} \in \mathbb{R}^d$ and let $q \in \mathbb{R}^d$ be a query vector. The *attention mechanism* is defined as

$$\text{att}(\boldsymbol{x}, q) = W_o \sum_{t=1}^{T} \alpha^{(t)}(q) W_v x^{(t)}, \quad (2)$$
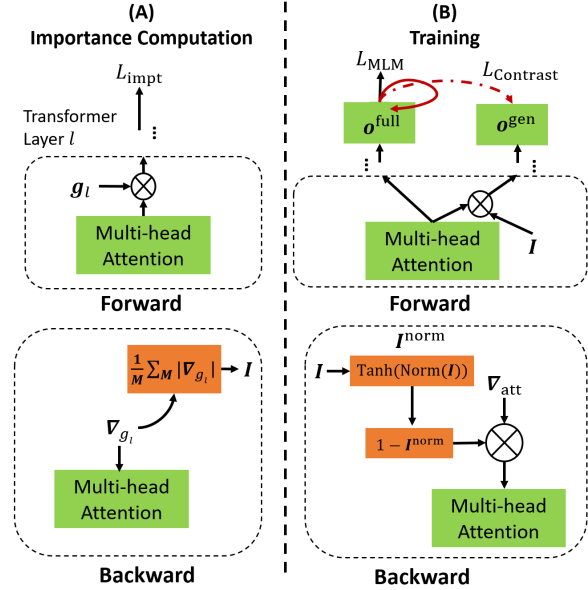

Figure 1: Illustration of DGA. **(A)** shows the importance computation. This is done by adding a gate vector $\boldsymbol{g}_l$ multiplying with the multi-head attention (Eq. 5) and averaging its training gradients (Eq. 6). **(B)** shows DGA training. In backward pass, attention heads are soft-masked based on their importance $\boldsymbol{I}$ (Eqs. 9 and 10) to try to preserve the general knowledge in the LM as much as possible. In forward pass, the added gate vector is removed except for feature learning in the contrastive loss. The contrastive loss is computed by contrasting the general knowledge with importance ($\boldsymbol{o}^{\text{gen}}$ in Eq. 12) applied and the full knowledge without applying the importance ($\boldsymbol{o}^{\text{full}}$ in Eq. 14). The final objective of DGA consists of MLM loss and contrastive loss. Note that we omit the details of other parts of Transformer and only focus on the multi-head attention mechanism.

where

$$\alpha^{(t)}(q) = \text{softmax}(\frac{q^T W_q^T W_k x^{(t)}}{\sqrt{d}}). \quad (3)$$

The projection matrices $W_o, W_v, W_q, W_k \in \mathbb{R}^{d \times d}$ are learnable parameters. The query vector is from the same sequence as $\boldsymbol{x}$ in self-attention. A Transformer contains L identical layers. For layer l, H_l different attention heads are applied in parallel to enable the Transformer to be trained on more data. Simply put, *multi-head attention* (mhatt) is the simultaneous application of multiple attention heads in a single Transformer architecture. They are then applied in parallel to obtain multi-head attention.[5]

$$\text{mhatt}_l(\boldsymbol{x}, q) = \sum_{h=1}^{H_l} \text{att}_{lh}(\boldsymbol{x}, q), \quad (4)$$

[5]We follow the notation in (Michel et al., 2019), where the notation in Eq. 4 is equivalent to the "concatenation" formulation in (Vaswani et al., 2017).

where h indicates the hth attention head. Note that the input x is different in each layer since the input of a given layer is the output of last layer. To ease the notation, we use the input x for all layers.

Head importance. Researchers have found that not all attention heads are important (Michel et al., 2019). We introduce a *gate vector*, g_l, where each cell is a *gate variable*, g_{lh}, to the attention head summation for detecting the importance of attention heads. The resulting importance scores are used to soft-mask the heads in DA-training.

$$\text{gmhatt}_l(\boldsymbol{x}, q) = \sum_{h=1}^{H_l} g_{lh}, \otimes \text{att}_{lh}(\boldsymbol{x}, q) \quad (5)$$

where $\otimes$ is the element-wise multiplication. A gradient-based head importance detection method is proposed in (Michel et al., 2019). Given a dataset $D = \{(\boldsymbol{y}_m, \boldsymbol{x}_m)\}_{m=1}^M$ of M samples ($\boldsymbol{y}_m$ is the label of $\boldsymbol{x}_m$ as Michel et al. (2019) worked on supervised learning), the importance of a head is estimated with a gradient-based proxy score

$$I_{lh} = \frac{1}{M} \sum_{m=1}^M |\nabla_{g_{lh}}|, \quad (6)$$

where $\nabla_{g_{lh}}$ is the gradient of gate variable g_{lh},

$$\nabla_{g_{lh}} = \frac{\partial \mathcal{L}_{\text{impt}}(\boldsymbol{y}_m, \boldsymbol{x}_m)}{\partial g_{lh}}, \quad (7)$$

where $\mathcal{L}_{\text{impt}}$ is a task-specific/domain-specific loss function. The gradient can be used as the importance score because changing g_{lh} is liable to have a large effect on the model if I_{lh} has a high value.

Although Eq. 6 offers a way to compute the importance of attention heads *w.r.t. a given loss* $\mathcal{L}_{\text{impt}}$, we are unable to directly apply it: If we use the domain data at hand and the MLM loss as $\mathcal{L}_{\text{impt}}$, $\nabla_{g_{lh}}$ only indicates the importance score for *domain-specific* knowledge. However, our goal is to estimate the attention heads importance for the *general knowledge* in LM which requires the data used in training the LM to compute the $\mathcal{L}_{\text{impt}}$. In practice, such data is not accessible to users of the LM. Further, label is needed in Eq. 6 but our domain corpus is unlabeled in DA-training. To address these issues, we propose to compute a *proxy KL-divergence loss* for $\mathcal{L}_{\text{impt}}$.

Proxy KL-divergence loss. We need a proxy for $\mathcal{L}_{\text{impt}}$ such that its gradient ($\nabla_{g_{lh}}$) can be used to compute head importance without using the LM's original pre-training data. We propose to use model

robustness as the proxy, i.e., we try to detect heads that are important for LM's *robustness*. Its gradient, $\nabla_{g_{lh}}$, then indicates the robustness and thus the importance to the LM model. Our rationale is as follows: If an I_{lh} (the average of $|\nabla_{g_{lh}}|$, see Eq. 6) has a high value, it indicates that it is important to the LM's robustness because its change can cause the LM to change a great deal. It is thus an important head to the LM. In contrast, if I_{lh} has a small value, it is a less or not important head to the LM.

To compute the *robustness* of the LM, we take a subset (a hyper-parameter) of the target domain data $\{\boldsymbol{x}_m^{\text{sub}}\}$ (no label in DA-training) and input $\boldsymbol{x}_m^{\text{sub}}$ twice to the LM and compute the KL-divergence of the two resulting representations,

$$\mathcal{L}_{\text{impt}} = \text{KL}(f_1(\boldsymbol{x}_m^{\text{sub}}), f_2(\boldsymbol{x}_m^{\text{sub}})), \quad (8)$$

where f_1 and f_2 are the LM with different dropout masks. Note that we don't need to add any additional dropouts to implement f because independently sampled *dropout masks* are used as input in the Transformer. In training a Transformer, there are dropout masks placed on fully-connected layers and attention probabilities. Thus, simply feeding the same input to the Transformer twice will get two representations with different dropout masks. Since dropout is similar to adding noise, the difference between the two representations can be regarded as the *robustness* of the Transformer model. Figure 1 (A) shows how we compute the importance of each attention head using the gradient of the gate vector g_l.

Soft-masking attention heads in DA-training. Recall we want to preserve the general knowledge in the LM during DA-training using head importance I_{lh}. Given the attention head $\text{att}(\boldsymbol{x}, q)$ and DA-training loss $\mathcal{L}_{\text{DA-train}}$ (typically the MLM loss; we also propose an additional loss in Sec. 3.2), we can "soft mask" its corresponding gradient ($\nabla_{\text{att}_{lh}}$ [6]) using the head importance value I_{lh},

$$\nabla'_{\text{att}_{lh}} = (1 - I_{lh}^{\text{norm}}) \otimes \nabla_{\text{att}_{lh}}, \quad (9)$$

where I_{lh}^{norm} is from I_{lh} via normalization

$$I_{lh}^{\text{norm}} = |\text{Tanh}(\text{Normalize}(I_{lh}))|. \quad (10)$$

Normalize makes the I_{lh} have a mean of 0 and standard deviation of 1. Absolute value of Tanh

---

[6]$\nabla_{\text{att}_{lh}}$ indicates the gradient for attention head $\text{att}_{lh}(\boldsymbol{x}, q)$, distinguished from $\nabla_{g_{lh}}$ in Eq. 6 which is the gradient for the gate variable g_{lh}

ensures that I_{lh} takes values in the interval $[0,1]$. Eq. 9 means to constrain the gradient of the corresponding head $\text{att}_{lh}(\boldsymbol{x},q)$ by element-wise multiplying one minus the head importance to the gradient. It is "soft-masking" because I_{lh} is a real number in $[0,1]$ (instead of binary $\{0,1\}$), which gives the model the flexibility to adjust the attention head. This is useful because although some heads are important to the LM, they may conflict with the knowledge in the target domain and thus need adjusting. Also note that the soft masks here affect only the backward pass and are not used in forward pass (so that forward pass can use the full network and encourage maximal integration of pre-trained general and domain-specific knowledge) except for feature learning using contrastive learning (see below). Figure 1 (B) shows that attention heads are soft-masked during training.

3.2 Contrasting General and Full Knowledge

We now present how to integrate the general knowledge in the LM and the domain-specific knowledge in the target domain by contrasting the general knowledge and the full knowledge (both general and domain-specific). We first introduce how we obtain such knowledge from the LM for the input $\boldsymbol{x}$, and then discuss how we contrast them.

Obtaining the general knowledge for the input sequence $\boldsymbol{x}$ from the LM is by extracting the representation of combining the attention heads and their importance scores (I_{lh}^{norm} in Eq. 10) in the forward pass. The intuition is that since the importance scores show how important each attention head is to the general knowledge, the resulting representation reflects the main general knowledge used by $\boldsymbol{x}$. Formally, we plug I_{lh}^{norm} (soft-masks) as the gate variable g_{lh} in Eq. 5,

$$\text{gmhatt}_l^{\text{gen}}(\boldsymbol{x},q) = \sum_{h=1}^{H_l} I_{lh}^{\text{norm}} \otimes \text{att}_{lh}(\boldsymbol{x},q). \quad (11)$$

Given the attention heads for general knowledge, we can plug it into the whole Transformer to obtain the final general knowledge (taking the average of each token's output in the input sequence).

$$\boldsymbol{o}^{\text{gen}} = \text{Transformer}(\text{gmhatt}^{\text{gen}}(\boldsymbol{x},q)). \quad (12)$$

(See $\boldsymbol{o}^{\text{gen}}$ also in Figure 1 (B)).

Obtaining the full (both general and domain-specific) knowledge in $\boldsymbol{x}$ is similar. The only difference is that we extract the representation of $\boldsymbol{x}$

without applying the importance (soft-masks) on attention heads in the forward pass,

$$\text{gmhatt}_l^{\text{full}}(\boldsymbol{x},q) = \sum_{h=1}^{H_l} \text{att}_{lh}(\boldsymbol{x},q). \quad (13)$$

Similarly, we can plug it into the Transformer,

$$\boldsymbol{o}^{\text{full}} = \text{Transformer}(\text{gmhatt}^{\text{full}}(\boldsymbol{x},q)). \quad (14)$$

(See $\boldsymbol{o}^{\text{full}}$ also in Figure 1 (B)). Note that it is possible to use $(1 - I_{lh}^{\text{norm}})$ as the importance of domain-specific knowledge and contrast it with the general knowledge. However, this produces poorer results (see Table 3) as explained in footnote 4.

Contrasting general and full knowledge. It is known that contrastive learning helps learn a good isotropic representation that is good for downstream tasks, with the help of positive and negative instances. We contrast the general ($\boldsymbol{o}^{\text{gen}}$) and full ($\boldsymbol{o}^{\text{full}}$) representations (as positive and negative instances) for the same input $\boldsymbol{x}$ to make them different, which encourages the learning of domain-specific knowledge in $\boldsymbol{o}^{\text{full}}$ that is not already in the general knowledge and yet related to and integrated with the general knowledge ($\boldsymbol{o}^{\text{gen}}$) of the input.

We construct contrastive instances as follows: for an input $\boldsymbol{x}_m$, three contrastive instances are produced. Anchor $\boldsymbol{o}_m$ and positive instance $\boldsymbol{o}_m^+$ are both full knowledge from Eq. 14, obtained based on two independently sampled dropout masks in the Transformer (recall that this can be achieved by inputting $\boldsymbol{x}_m$ twice (see Sec. 3.1). We regard $\boldsymbol{o}_m^+$ and $\boldsymbol{o}_m$ as positive instances because the dropout noise has been shown to be good positive instances for improving alignment in training sentence embedding (Gao et al., 2021a). Negative instance $\boldsymbol{o}_m^-$ is the general knowledge for $\boldsymbol{x}_m$ from the LM obtained via Eq. 12. With $\boldsymbol{o}_m$, $\boldsymbol{o}_m^+$, and $\boldsymbol{o}_m^-$, our contrastive loss is ($\text{sim}(\cdot)$ is the cosine similarity),

$$\mathcal{L}_{\text{contrast}} = -\log \frac{e^{\text{sim}(\boldsymbol{o}_m, \boldsymbol{o}_m^+)/\tau}}{\sum_{j=1}^{N}(e^{\text{sim}(\boldsymbol{o}_m, \boldsymbol{o}_j^+)/\tau} + e^{\text{sim}(\boldsymbol{o}_m, \boldsymbol{o}_j^-)/\tau})}. \quad (15)$$

Compared to Eq. 1, the second term is added in the denominator, i.e., general knowledge representations as additional negative samples/instances. Figure 1 (B) shows a red arrow pointed from $\boldsymbol{o}^{\text{full}}$ to itself, indicating the positive instances are from inputting twice. The dashed red arrow pointing to $\boldsymbol{o}^{\text{gen}}$ indicates the negative instances contrasting the specialized and general knowledge.

Unlabeled Domain Datasets			End-Task Classification Datasets				
Source	Dataset	Size	Dataset	Task	#Training	#Testing	#Classes
Reviews	Yelp Restaurant	758MB	Restaurant	Aspect Sentiment Classification (ASC)	3,452	1,120	3
	Amazon Phone	724MB	Phone	Aspect Sentiment Classification (ASC)	239	553	2
	Amazon Camera	319MB	Camera	Aspect Sentiment Classification (ASC)	230	626	2
Academic Papers	ACL Papers	867MB	ACL	Citation Intent Classification	1,520	421	6
	AI Papers	507MB	AI	Relation Classification	2,260	2,388	7
	PubMed Papers	989MB	PubMed	Chemical-protein Interaction Prediction	2,667	7,398	13

Table 1: Statistics for domain post-training datasets and end task supervised classification datasets (more detail of each task is given in Appendix A).

3.3 DGA Objectives

DGA is a pipelined model: First, a subset of the domain data is used to estimate the attention head importance (I_{lh} in Sec. 3.1). Second, given the attention head importance, we compute the final domain-adaptive loss by combining the conventional Masked Language Model (MLM) loss (include the proposed soft-masking for general knowledge) and the proposed contrastive loss:

$$\mathcal{L}_{\text{DA-train}} = \mathcal{L}_{\text{MLM}} + \lambda_1 \mathcal{L}_{\text{contrast}}, \qquad (16)$$

where λ_1 is the hyper-parameter to adjust the impact of the added term.

4 Experiments

We follow the experiment setup in (Gururangan et al., 2020). RoBERTa (Liu et al., 2019)[7] is used as the LM. In each experiment, we first DA-train the LM and then fine-tune it on the end-task. The final evaluation is based on the end-task results.

4.1 Datasets and Baselines

Datasets: Table 1 shows the statistics of the *unlabeled domain datasets* for DA-training and their corresponding *end-task classification datasets*. We use 6 *unlabeled domain datasets*:[8] 3 of them are about reviews: *Yelp Restaurant* (Xu et al., 2019a), *Amazon Phone* (Ni et al., 2019), *Amazon Camera* (Ni et al., 2019); 3 of them are academic papers: *ACL Papers* (Lo et al., 2020), *AI Papers* (Lo et al., 2020), and *PubMed Papers*[9]. Each unlabeled domain dataset has a corresponding *end-task classifi-*

---

[7]https://huggingface.co/roberta-base

[8]We down-sampled the *PubMed* due to its huge original size. In general, our datasets are much smaller comparing to previous work (Gururangan et al., 2020) (which used more than 11GB of data for each domain). Our experiments showed that a smaller dataset is sufficient and more data does not help. It also requires much less computation resource.

[9]https://pubmed.ncbi.nlm.nih.gov/

cation dataset[10]: *Restaurant*[11] (Xu et al., 2019a), *Phone* (Ding et al., 2008; Hu and Liu, 2004), *Camera* (Ding et al., 2008; Hu and Liu, 2004)[12], *ACL* (ACL-ARC in Jurgens et al. (2018)), *AI* (SCIERC in Luan et al. (2018)), and PubMed (CHEMPORT in Kringelum et al. (2016)).

Baselines. We consider 10 baselines.

(1). **Non-DA-training (RoBERTa)** (Liu et al., 2019) uses the original RoBERTa for the end-task fine-tuning without any DA-training.

(2). **DA-training using masked language model loss (MLM)** is the existing DA-training method. To our knowledge, existing DA-training systems are all based on the MLM loss.

(3). **DA-training using adapter-tuning (MLM (Adapter))** adds adapter layers between layers of Transformer for DA-training. An adapter (Houlsby et al., 2019) has two fully connected layers and a skip connection. During DA-training, the Transformer is fixed, only the adapters are trained. The bottleneck (adapter) size is set to 64 (Houlsby et al., 2019). During end-task fine-tuning, both RoBERTa and adapters are trainable for fair comparison.

(4). **DA-training using prompt-tuning (MLM (Prompt))** (Lester et al., 2021) adds a sequence of prompt tokens to the end of the original sequence. In DA-training, RoBERTa (the LM) is fixed and only the prompt tokens are trained. In end-task fine-tuning, both LM and the trained prompt are trainable. We initialize 100 tokens and set the learning rate of the prompt token to 0.3 in DA-training, following the setting in Lester et al. (2021).

(5). **Knowledge distillation (MLM+KD)** (Hin-

---

[10]Note that our results are different from those presented in Table 5 of (Gururangan et al., 2020) because we observe very high variances due to very small original test sets and thus re-partition the training and test set (by enlarging the test set and reducing the training set slightly)

[11]To be consistent with existing research (Tang et al., 2016), examples with conflict polarities (both positive and negative sentiments are expressed about an aspect term) are not used.

[12]Note that Ding et al. (2008) and Hu and Liu (2004) contain 9 and 5 domains, respectively. We extract those domains related to "Phone" and "Camera" from them.

ton et al., 2015) minimizes the representational deviation between the general knowledge in LM and the specialized knowledge in DA-training. We compute the KL divergence between the representations (the output before the masked language model prediction head) of each word of the two models (LM and DA-trained) as the distillation loss.

(6). **Adapted distillation through attention (MLM+AdaptedDeiT)** is derived from DeiT (Touvron et al., 2021), a distillation method for visual Transformer (ViT) (Dosovitskiy et al., 2020). We adapt DeiT to a text-based and unsupervised model by distilling the LM representation[13] to the added distillation token and change ViT to RoBERTa.

(7, 8). **DA-training using sequence-level contrastive learning (MLM+SimCSE and MLM+InfoWord)**. SimCSE is a contrastive learning method for sentence embedding (Gao et al., 2021a). We use its unsupervised version where positive samples are from the same input with different dropout masks and negative samples are other instances in the same batch. InfoWord (Kong et al., 2020) is another contrastive learning method contrasts the span-level local representation and sequence-level global representation.

(9, 10). **DA-training using token-aware contrastive learning (MLM+TaCL and MLM+TaCO)**. TaCL (Su et al., 2021) and TaCO (Fu et al., 2022) are two recent methods to improve BERT pre-training with token-aware contrastive loss.[14] We change the backbone to RoBERTa for fair comparison.

4.2 Implementation Detail

Architecture. We adopt RoBERTa$_{\text{BASE}}$ as our backbone LM (12 layers and 12 attention heads in each layer). A masked language model head is applied for DA-training. The end-task fine-tuning of RoBERTa follows the standard practice. For the three ASC tasks (see Table 1), we adopt the ASC formulation in (Xu et al., 2019a), where the aspect (e.g., "*sound*") and review sentence (e.g., "*The sound is great*") are concatenated via </s>.

Hyperparameters. Unless otherwise stated, the same hyper-parameters are used in all experiments. The maximum input length is set to 164 which is long enough for all datasets. Adam optimizer is

used for both DA-training and end-task fine-tuning. The max sequence length is set to 164, which is long enough for our end-tasks and only needs moderate computational resources.

Domain-adaptive pre-training (DA-training). The learning rate is set to 1e-4 and batch size is 256. We train 2.5K steps for each domain, roughly a full pass through the domain data, following (Gururangan et al., 2020; Xu et al., 2019a). The subset of data $\{x_m^{\text{sub}}\}$ for computing $\mathcal{L}_{\text{impt}}$ to determine head importance in Sec. 3.1 is set to 1.64 Million tokens, which is sufficient in our experiments. λ_1 in Eq. 16 is set to 1 and τ in Eq. 15 is set to 0.05.

End-task fine-tuning. The learning rate is set to 1e-5 and batch size to 16. We train on end-task fine-tuning datasets for 5 epochs for Restaurant; 10 epochs for ACL, AI and PubMed; and 15 epochs for Phone and Camera. We simply take the results for the last epoch as we empirically found that the above number of epochs gives us stable and convergence results.

4.3 Evaluation Results and Ablation Study

We report the end-task results of the 10 baselines on the 6 datasets in Table 2.

Superiority of DGA. Our DGA consistently outperforms all baselines. Thanks to the proposed more informed adaptation, DGA improves over the widely used *traditional DA-training baseline* MLM. We also see that MLM markedly outperforms RoBERTa (non-DA-training) on average (see the last column). We discuss more observations about the results bellow.

(1). Training the entire LM in DGA helps achieve much better results. Using adapter (MLM (adapter)) and prompt (MLM (prompt)) have mixed results. This is because adapter and prompt do not have sufficient trainable parameters, which are also randomly initialized and can be difficult to train.

(2). DGA is also better than distillation-based systems: MLM+AdaptedDeiT and MLM+KD, which try to preserve the past knowledge. This is not surprising because the goal of DA-training is not simply preserving the previous knowledge but also to adapt/change it as needed to suit the target domain. DGA is specifically designed for this with soft-masking and contrasting of knowledge.

(3). The contrastive learning in DGA is more effective than the other contrastive alternatives (MLM+SimCSE, MLM+TaCL, MLM+TaCO and MLM+InfoWord). This indicates contrasting the

---

[13]We take the average of its token's output as sequence's representation. The same for SimCSE baseline.

[14]TaCL and TaCO are not a DA-training model. It pre-trains an LM to improve it using the same data as that for training the LM. We switch the data to our target domain data.

| Domain | Camera | | Phone | | Resturant | | AI | | ACL | | PubMed | Avg |
Model	MF1	Acc.	MF1	Acc.	MF1	Acc.	MF1	Acc.	MF1	Acc.	Micro-F1	
RoBERTa	78.82	87.03	83.75	86.08	79.81	87.00	60.98	71.85	66.11	71.26	72.38	73.64
MLM	84.39	89.90	82.59	85.50	80.84	87.68	68.97	75.95	68.75	73.44	72.84	76.40
MLM (Adapter)	83.62	89.23	82.71	85.35	80.19	87.14	60.55	71.38	68.87	72.92	71.68	74.60
MLM (Prompt)	85.52	90.38	84.17	86.53	79.00	86.45	61.47	72.36	66.66	71.35	73.09	74.98
MLM+KD	82.79	89.30	80.08	83.33	80.40	87.25	67.76	75.46	68.19	72.73	72.35	75.26
MLM+AdaptedDeiT	86.86	91.37	83.08	85.64	79.70	86.84	69.72	76.83	69.11	73.35	72.69	76.86
MLM+SimCSE	84.91	90.35	83.46	86.08	80.88	87.59	69.10	76.25	69.89	74.30	72.77	76.84
MLM+TaCL	81.98	88.88	81.87	84.92	81.12	87.50	64.04	73.18	63.18	70.31	69.46	73.61
MLM+TaCO	84.50	90.22	82.63	85.32	79.27	86.68	59.73	71.22	63.66	70.36	72.38	73.69
MLM+InfoWord	87.95	91.92	84.58	86.84	81.24	87.82	68.29	75.92	68.58	73.68	73.21	77.31
DGA	**88.52**	**92.49**	**85.47**	**87.45**	**81.83**	**88.20**	**71.99**	**78.06**	**71.01**	**74.73**	**73.65**	**78.74**

Table 2: We report the macro-F1 (MF1) and accuracy results for all datasets, except for CHEMPORT in the PubMed domain, for which we use micro-F1 following Gururangan et al. (2020); Dery et al. (2021); Beltagy et al. (2019). The results are averages of 5 random seeds (the standard deviation is reported in Appendix B). The average column (Avg) is the average over the MF1 (or Micro-F1 for PubMed) for all datasets.

general and full knowledge for knowledge integration is important.

Effectiveness of the proxy KL-divergence loss. We use the proposed *proxy KL-divergence loss* to compute the head importance to identify the general language knowledge in the LM without using the LM's original pre-training data (Sec. 3.1).

For evaluation, we are interested in how good the proxy is. Since we don't have the data that pre-trains RoBERTa, it is not obvious how to assess the quality of the proxy directly. Here, we provide some indirect evidences to show the effectiveness of the proxy for computing the importance of units to the general knowledge in the LM.

We conduct a separate experiment to compare the attention heads' importance score vectors after applying the proxy using the data from different domains. For each domain i, we compare its importance vector with the importance vector of every other domain, and then average the cosine similarities to get the value for domain i. We get 0.92 for Restaurant, the same 0.91 for ACL, AI, and Phone, 0.89 for PubMed and 0.92 for Camera. We see that different domains give similar importance values, which indirectly show that our proxy can identify the common general knowledge.

We also compute the importance score distributions of the proxy. For each of the 6 domains, after applying the proxy, around 20% of the attention heads are heavily protected ($0.8 \leq I_{lh}^{norm} \leq 1.0$) and another 20% moderately protected ($0.6 \leq I_{lh}^{norm} < 0.8$), which indicate the general knowledge. While Phone, AI, Camera and Restaurant share a similar distribution, ACL and PubMed protect slightly less. This is understandable as PubMed

and ACL (medical or NLP publications) are probably less common than the other domains and the general knowledge in the LM covers them less.

Ablation study. To better understand DGA, We want to know (1) whether constraining the neurons in other layers are helpful (the proposed DGA only constrains the attention heads), and (2) where the gain of DGA is from. To answer (1), we constrain the training of different layers in a standard Transformer. In Table 3 (rows 3-5), "**H**", "**I**", and "**O**" refer to attention head, intermediate layer, output layer in a standard Transformer layer, respectively. "**E**" refers to the embedding layers. The brackets with combination of "H, I, O, E" indicate the location we apply the soft-masking (DGA only applies soft-masking in the attention head). We can see their results are similar or worse than DGA, implying that attention heads are more indicative of important knowledge. To answer (2), we conduct the following ablation experiments: (i) **DGA (w/o contrast)**, without the contrastive loss, but only soft-masking the backward pass according to the attention head importance. (ii) **DGA (random masking)** with randomly generated attention head importance scores and using them to do soft-masking and contrastive learning. (iii) **Ensemble (LM+MLM)** performs the end-task fine-tuning on both the MLM DA-trained RoBERTa (conventional DA-training) and the original RoBERTa (LM) by concatenating their outputs and taking the average. (iv) **DGA (domain-specific)** refers to the variant that contrasts domain-specific and general knowledge (see Sec. 3.2).[15]

---

[15]We don't have DGA(w/o soft-masking) because our con-

| Domain | Camera | | Phone | | Resturant | | AI | | ACL | | PubMed | Avg |
Model	MF1	Acc.	MF1	Acc.	MF1	Acc.	MF1	Acc.	MF1	Acc.	Micro-F1	
RoBERTa	78.82	87.03	83.75	86.08	79.81	87.00	60.98	71.85	66.11	71.26	72.38	73.64
MLM	84.39	89.90	82.59	85.50	80.84	87.68	68.97	75.95	68.75	73.44	72.84	76.40
DGA (H, I)	86.79	91.60	84.21	86.40	81.32	87.91	71.07	77.36	69.50	73.82	73.34	77.71
DGA (H, I, O)	88.04	92.01	85.85	87.63	81.45	87.79	71.54	77.61	70.52	74.58	73.10	78.42
DGA (H, I, O, E)	87.05	91.60	83.74	86.11	80.64	87.61	72.64	78.17	71.24	74.96	73.54	78.14
DGA (w/o contrast)	86.19	90.89	84.48	86.65	81.70	87.93	68.25	75.49	69.31	73.73	72.72	77.11
DGA (random mask)	82.07	89.30	83.86	86.33	80.60	87.52	69.51	76.64	69.59	73.73	72.92	76.43
Ensemble (LM+MLM)	85.22	90.64	85.15	87.23	79.86	86.98	65.10	74.43	68.56	73.44	72.60	76.08
DGA (domain-specific)	88.06	92.04	83.45	85.82	81.72	87.90	68.00	75.57	70.91	75.06	73.17	77.55
DGA	**88.52**	**92.49**	**85.47**	**87.45**	**81.83**	**88.20**	**71.99**	**78.06**	**71.01**	**74.73**	**73.65**	**78.74**

Table 3: Ablation results - averages of 5 random seeds. The standard deviations are reported in Appendix B.

Table 3 shows that the full DGA always gives the best result, indicating every component contributes. Additional observations are as follows:

(1) DGA's gain is partially from the novel soft-masking: we can see that on average, DGA (w/o contrast) outperforms conventional DA-training (MLM). Besides, our gradient-based mask is informative: we can see DGA (random mask) is worse than DGA (w/o contrast) on all datasets. DGA (w/o contrast) is even better than Ensemble, which directly combines the information given by both the original LM and the traditional DA-trained model during end-task fine-tuning

(2) Besides soft-masking, contrasting the general and full knowledge also helps. We can see DGA outperforms DGA (w/o contrast) and DGA (domain-specific) in all datasets.

5 Conclusion

This paper argued that an effective DA-training method should effectively integrate the target domain knowledge to the general knowledge in the LM. Existing approaches do not explicitly do this. This paper proposed a novel method DGA to achieve it (1) by estimating the attention heads importance in LM and using the importance scores to soft-mask the attention heads in DA-training to preserve the important knowledge in LM as much as possible, and (2) by contrasting the general and the full knowledge. Extensive experiment results demonstrated the effectiveness of the proposed approach DGA.

6 Limitations

While effective, DGA has some limitations. First, the main focus of DGA is to adapt an LM to a

given target domain. It does not consider the generalization to other domains. For example, it will be interesting to incrementally or continually adapt an LM to more and more domains to make the LM more useful. Second, the importance of parameters for general knowledge in the LM is computed using a proxy method based on model robustness. Although it is quite effective, it is interesting to explore other approaches to further improve it. We will work on these in our future work as specializing and improving an LM is an important problem.

Acknowledgments

References

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*.

Tuhin Chakrabarty, Christopher Hidey, and Kathleen McKeown. 2019. Imho fine-tuning improves claim detection. *arXiv preprint arXiv:1905.07000*.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020a. The lottery ticket hypothesis for pre-trained bert networks. *Advances in neural information processing systems*, 33:15834–15846.

trastive learning relies on soft-masking. If removed, contrastive loss will not have the additional negative samples and our DGA becomes MLM+SimCSE.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Zhiyuan Chen and Bing Liu. 2018. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207.

Lucio M Dery, Paul Michel, Ameet Talwalkar, and Graham Neubig. 2021. Should we be pre-training? an argument for end-task aware training as an alternative. *arXiv preprint arXiv:2109.07437*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*.

Zhiyi Fu, Wangchunshu Zhou, Jingjing Xu, Hao Zhou, and Lei Li. 2022. Contextual representation learning beyond masked language modeling. In *ACL*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021a. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Yang Gao, Nicolo Colombo, and Wei Wang. 2021b. Adapting by pruning: A case study on bert. *arXiv preprint arXiv:2105.03343*.

Yiwen Guo, Anbang Yao, and Yurong Chen. 2016. Dynamic network surgery for efficient dnns. *Advances in neural information processing systems*, 29.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *ACL*.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

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 *ICML*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD*.

David Jurgens, Srijan Kumar, Raine Hoover, Daniel A. McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *TACL*.

Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. 2020. A mutual information maximization perspective of language representation learning. In *ICLR*.

Jens Kringelum, Sonny Kim Kjaerulff, Søren Brunak, Ole Lund, Tudor I Oprea, and Olivier Taboureau. 2016. Chemprot-3.0: a global chemical biology diseases mapping. *Database*, 2016.

Cheng-I Jeff Lai, Yang Zhang, Alexander H Liu, Shiyu Chang, Yi-Lun Liao, Yung-Sung Chuang, Kaizhi Qian, Sameer Khurana, David Cox, and Jim Glass. 2021. Parp: Prune, adjust and re-prune for self-supervised speech recognition. *NeurIPS*, 34.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *EMNLP*.

Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. 2021. Differentiable subset pruning of transformer heads. *Transactions of the Association for Computational Linguistics*, 9:1442–1459.

Zi Lin, Jeremiah Zhe Liu, Zi Yang, Nan Hua, and Dan Roth. 2020. Pruning redundant mappings in transformer models via spectral-normalized identity prior. *arXiv preprint arXiv:2010.01791*.

Bing Liu. 2015. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.

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*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S. Weld. 2020. S2ORC: the semantic scholar open research corpus. In *ACL*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *ACL*.

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

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP*, pages 188–197. Association for Computational Linguistics.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2020. On the effect of dropping layers of pre-trained transformer models. *arXiv preprint arXiv:2004.03844*.

Yixuan Su, Fangyu Liu, Zaiqiao Meng, Lei Shu, Ehsan Shareghi, and Nigel Collier. 2021. Tacl: Improving bert pre-training with token-aware contrastive learning. *arXiv preprint arXiv:2111.04198*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *EMNLP*.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR.

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 *NeurIPS*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019a. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *NAACL-HLT*.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2019b. Review conversational reading comprehension. *arXiv preprint arXiv:1902.00821*.

Domain	Camera		Phone		Resturant		AI		ACL		PubMed
Model	MF1	Acc.	MF1	Acc.	MF1	Acc.	MF1	Acc.	MF1	Acc.	Micro-F1
RoBERTa	±0.0403	±0.0179	±0.0210	±0.0154	±0.0117	±0.0049	±0.0646	±0.0347	±0.0192	±0.0096	±0.0071
MLM	±0.0479	±0.0298	±0.0165	±0.0103	±0.0096	±0.0056	±0.0117	±0.0086	±0.0218	±0.0118	±0.0035
MLM (adapter)	±0.0165	±0.0110	±0.0265	±0.0181	±0.0102	±0.0068	±0.0551	±0.0288	±0.0142	±0.0099	±0.0055
MLM (prompt)	±0.0243	±0.0138	±0.0126	±0.0087	±0.0060	±0.0035	±0.0301	±0.0124	±0.0068	±0.0108	±0.0028
MLM+KD	±0.0295	±0.0158	±0.0320	±0.0230	±0.0099	±0.0070	±0.0345	±0.0224	±0.0292	±0.0155	±0.0093
MLM+AdaptedDeiT	±0.0187	±0.0122	±0.0160	±0.0101	±0.0048	±0.0022	±0.0250	±0.0179	±0.0065	±0.0079	±0.0086
MLM+SimCSE	±0.0114	±0.0077	±0.0098	±0.0065	±0.0029	±0.0016	±0.0086	±0.0056	±0.0054	±0.0071	±0.0027
MLM+TaCL	±0.0218	±0.0103	±0.0230	±0.0159	±0.0105	±0.0059	±0.0275	±0.0156	±0.0713	±0.0394	±0.0118
MLM+TaCO	±0.0456	±0.0232	±0.0166	±0.0134	±0.0077	±0.0052	±0.0675	±0.0380	±0.0207	±0.0128	±0.0099
MLM+InfoWord	±0.0267	±0.0139	±0.0272	±0.0191	±0.0170	±0.0089	±0.0344	±0.0219	±0.0070	±0.0079	±0.0072
DGA	±0.0095	±0.0047	±0.0127	±0.0094	±0.0052	±0.0040	±0.0127	±0.0081	±0.0079	±0.0080	±0.0034

Table 4: Standard deviations of the corresponding metrics of the proposed DGA model and the baselines on the six experiments.

Domain	Camera		Phone		Resturant		AI		ACL		PubMed
Model	MF1	Acc.	MF1	Acc.	MF1	Acc.	MF1	Acc.	MF1	Acc.	Micro-F1
RoBERTa	±0.0403	±0.0179	±0.0210	±0.0154	±0.0117	±0.0049	±0.0646	±0.0347	±0.0192	±0.0096	±0.0071
MLM	±0.0479	±0.0298	±0.0165	±0.0103	±0.0096	±0.0056	±0.0117	±0.0086	±0.0218	±0.0118	±0.0035
DGA (H, I)	±0.0373	±0.0210	±0.0032	±0.0039	±0.0054	±0.0045	±0.0095	±0.0048	±0.0094	±0.0073	±0.0049
DGA (H, I, O)	±0.0167	±0.0092	±0.0182	±0.0155	±0.0055	±0.0033	±0.0093	±0.0075	±0.0080	±0.0070	±0.0056
DGA (H, I, O, E)	±0.0237	±0.0123	±0.0270	±0.0187	±0.0099	±0.0050	±0.0109	±0.0089	±0.0067	±0.0057	±0.0079
DGA (w/o contrast)	±0.0433	±0.0251	±0.0135	±0.0106	±0.0060	±0.0040	±0.0197	±0.0119	±0.0132	±0.0093	±0.0050
DGA (random mask)	±0.0879	±0.0413	±0.0335	±0.0235	±0.0096	±0.0044	±0.0153	±0.0090	±0.0105	±0.0059	±0.0052
Ensemble	±0.0332	±0.0178	±0.0199	±0.0139	±0.0035	±0.0031	±0.0236	±0.0103	±0.0061	±0.0028	±0.0046
DGA (domain-specific)	±0.0137	±0.0070	±0.0259	±0.0200	±0.0031	±0.0018	±0.0128	±0.0071	±0.0108	±0.0067	±0.0043
DGA	±0.0095	±0.0047	±0.0127	±0.0094	±0.0052	±0.0040	±0.0127	±0.0081	±0.0079	±0.0080	±0.0034

Table 5: Standard deviations of the corresponding metrics of the proposed DGA model and the ablation on the six experiments.

A Datasets Details

Table 2 in the main paper has given the number of examples in each dataset. Here we provide additional details about the 4 types of end-tasks.

(1) **(Phone, Camera and Restaurant) Aspect Sentiment Classification (ASC)** is defined as follows (Liu, 2015): given an aspect or product feature (e.g., *picture quality* in a camera review) and a review sentence containing the aspect in a domain or product category (e.g., camera), classify if the sentence expresses a positive, negative, or neutral (no opinion) sentiment or polarity about the aspect (for Phone and Camera, there are only negative and positive polarities in the data).

(2) **(ACL) Citation Intent Classification** is defined as follows: given a citing sentence (a sentence contains a citation), classify if the sentence expresses a citation function among "background", "motivation", "uses", "extension" and "comparison or contrast future".

(3) **(AI) Relation Classification** is defined as follows: given a within-sentence word sequence spans containing a pair of entities, classify if the span expresses a relation among "feature of", "conjunction", "evaluate for", "hyponym of", "used for", "part of" and "compare".

(4) **(PubMed) Chemical-protein Interaction Classification** is defined as follows: given a span containing a pair of chemical and protein, classify if the span expresses a chemical-protein interaction among "downregulator", "substrate", "indirect-upregulator", "indirect-downregulator", "agnonist", "activator", "product of", "agonist-activator", "inhibitor", "upregulator", "substrate product of", "agonist-inhibitor"and "antagonist".

B Standard Deviations

Table 4 reports the standard deviations of the corresponding results in Table 2 (in the main paper) of DGA and the considered baselines over 5 runs with random seeds. We can see the results of DGA are stable. Some baselines (e.g., RoBERTa in AI, MLM in Camera and MLM+TaCL in ACL) can have quite large standard deviations.

Table 5 reports the standard deviations of the corresponding results in Table 3 (in the main paper) of DGA and the considered baselines over 5 runs with random seeds. We can see the results of DGA are stable. Some baselines (e.g., DGA (random mask) and DGA (w/o contrast) in Camera) can have quite large standard deviations.