Taming Pre-trained Language Models with N-gram Representations for Low-Resource Domain Adaptation

Shizhe Diao[♦], **Ruijia Xu**[♦], **Hongjin Su**[♣], **Yilei Jiang**[♣]
Yan Song[♠♡], **Tong Zhang**[♦]
[♦]The Hong Kong University of Science and Technology
{sdiaoaa, rxuaq, tongzhang}@ust.hk
[♣]The Chinese University of Hong Kong
[♠]The Chinese University of Hong Kong (Shenzhen)
[♡]Shenzhen Research Institute of Big Data
songyan@cuhk.edu.cn

Abstract

Large pre-trained models such as BERT are known to improve different downstream NLP tasks, even when such a model is trained on a generic domain. Moreover, recent studies have shown that when large domain-specific corpora are available, continued pre-training on domain-specific data can further improve the performance of in-domain tasks. However, this practice requires significant domain-specific data and computational resources which may not always be available. In this paper, we aim to adapt a generic pretrained model with a relatively small amount of domain-specific data. We demonstrate that by explicitly incorporating the multi-granularity information of unseen and domain-specific words via the adaptation of (word based) n-grams, the performance of a generic pretrained model can be greatly improved. Specifically, we introduce a **T**ransformer-based **D**omain-aware **N**-gram **A**daptor, **T-DNA**, to effectively learn and incorporate the semantic representation of different combinations of words in the new domain. Experimental results illustrate the effectiveness of T-DNA on eight low-resource downstream tasks from four domains. We show that T-DNA is able to achieve significant improvements compared to existing methods on most tasks using limited data with lower computational costs. Moreover, further analyses demonstrate the importance and effectiveness of both unseen words and the information of different granularities.[1]

1 Introduction

Pre-trained language models have achieved great success and shown promise in various application scenarios across natural language understanding (Devlin et al., 2019; Liu et al., 2019; Tian et al., 2020a) and generation (Lewis et al., 2020; Zhang et al., 2020; Yang et al., 2020). Normally applying pre-trained language models to different applications follows a two-stage paradigm: pre-training on a large unlabeled corpus and then fine-tuning on a downstream task dataset. However, when there are domain gaps between pre-training and fine-tuning data, previous studies (Beltagy et al., 2019; Lee et al., 2020) have observed a performance drop caused by the incapability of generalization to new domains. Towards filling the gaps, the main research stream (Beltagy et al., 2019; Alsentzer et al., 2019; Huang et al., 2019; Lee et al., 2020) on adapting pre-trained language models starts from a generic model (e.g., BERT, RoBERTa) and then continues pre-training with similar objectives on a large-scale domain-specific corpus. However, without providing sufficient understanding of the reason for the performance drop during the domain shift, it is prone to failure of adaptation. Therefore, many aspects of continuous pre-training are expected to be enhanced. First, although generic pre-trained models offer better initialization for continuous pre-training models, it still costs considerable time (and money) that are beyond the reach of many institutions.[2] Second, it is clumsy to pre-train domain-specific models repeatedly for each domain on large-scale corpora.[3] Therefore, it is helpful to have an efficient and flexible method for being able to adapt pre-trained language models to different domains requiring limited resources.

Starting from the observed vocabulary mismatch problem (Gururangan et al., 2020), we further show empirically that the domain gap is largely caused by domain-specific n-grams.[4] Motivated by this find-

[1]Our code is available at https://github.com/shizhediao/T-DNA.

[2]For example, BioBERT (Lee et al., 2020), initialized by generic BERT, was trained on biomedical corpora for 23 days on eight NVIDIA V100 GPUs.

[3]For example, SciBERT (Beltagy et al., 2019) needs to be trained from scratch if one wants to use a domain-specific vocabulary (i.e., SciVocab in their paper).

[4]We explain it in detail in the following section.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 3336–3349
August 1–6, 2021.

ing, we propose a light-weight **T**ransformer-based **D**omain-aware **N**-gram **A**daptor (**T-DNA**) by incorporating n-gram representations to bridge the domain gap between source and target vocabulary. Specifically, the proposed model is able to explicitly learn and incorporate better representations of domain-specific words and phrases (in the form of n-grams) by the adaptor networks with only requiring small pieces of data. With this adaptor, once entering a new domain, one can choose to train the adaptor alone or train it with a Transformer-based backbone (e.g., BERT) together, where the joint training paradigm could provide more improvement. In addition, although it is designed for a low-resource setting, the adaptor is still able to work with enough data, which ensures its generalization ability in different scenarios.

Experimental results demonstrate that T-DNA significantly improves domain adaptation performance based on a generic pre-trained model and outperforms all baselines on eight classification tasks (on eight datasets). The results confirm that incorporating domain-specific n-grams with the proposed T-DNA is an effective and efficient solution to domain adaptation, showing that the information carried by larger text granularity is highly important for language processing across domains. Moreover, further analyses investigate the factors that may influence the performance of our model, such as the amount of available data, the training time cost and efficiency, and the granularity of domain-specific information, revealing the best way and setting for using the model.

2 The Motivation

As observed in Gururangan et al. (2020), the transfer gain of domain-specific pre-training becomes increasingly significant when the source and target domain are vastly dissimilar in terms of the vocabulary overlap. Motivated by this association between transfer gain and vocabulary distribution, we further investigate the shift of words and phrases across domains and attempt to alleviate the degradation of language models without large domain-specific corpora.

In particular, we start with a RoBERTa-base model from the generic domain and then fine-tune it on the IMDB (Maas et al., 2011) dataset. We investigate the outputs predicted by the [CLS] embedding on the IMDB development set and divide them into two categories: correct predictions (true positive/negative) and false predictions (false positive/false negative). To examine the vocabulary mismatch problem during the domain shift, we extract the top 1K most frequent n-grams[5] from these two categories respectively. We identify the n-grams not in the top 10K most frequent n-grams of source data[6] as domain-specific n-grams. As revealed in Figure 1, a larger proportion of domain-specific n-grams are captured when the model is misled to make wrong predictions, which suggests that the shifts in semantic meaning for both words and phrases might account for the domain shift. Furthermore, we conjecture that the representations of domain-specific n-grams are unreliable, which exacerbates the model degradation. While more details will be presented in §6.3, we briefly mention here that the tokens usually improperly attend to other tokens in the sentence but omit the most important words and phrases.

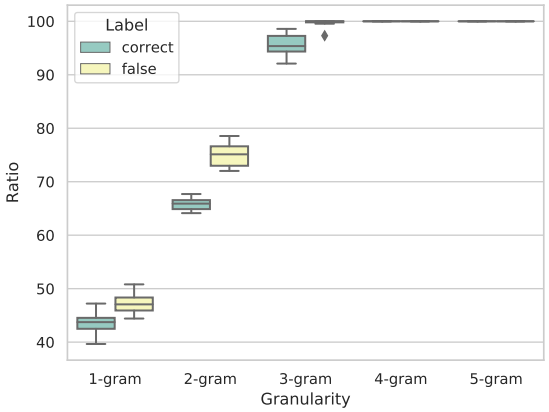

Figure 1: The proportion of domain-specific n-grams in correct predictions and false predictions over 10 different random seeds.

In light of this empirical evidence, we are motivated to design a framework to not only capture the domain-specific n-grams but also reliably embed them to extrapolate in the novel domain.

3 The T-DNA

Our approach follows the standard recipe of pre-training and fine-tuning a language model, which receives a sentence $\mathcal{X} = t_1 t_2 \cdots t_i \cdots t_T$ with t_i indicating the i-th token, and outputs the representation of each token. The overall architecture of our approach is shown in Figure 2. In the middle, a generic pre-trained encoder, such

[5]Here we set n to 5.

[6]We sample a subset from English Wikipedia.

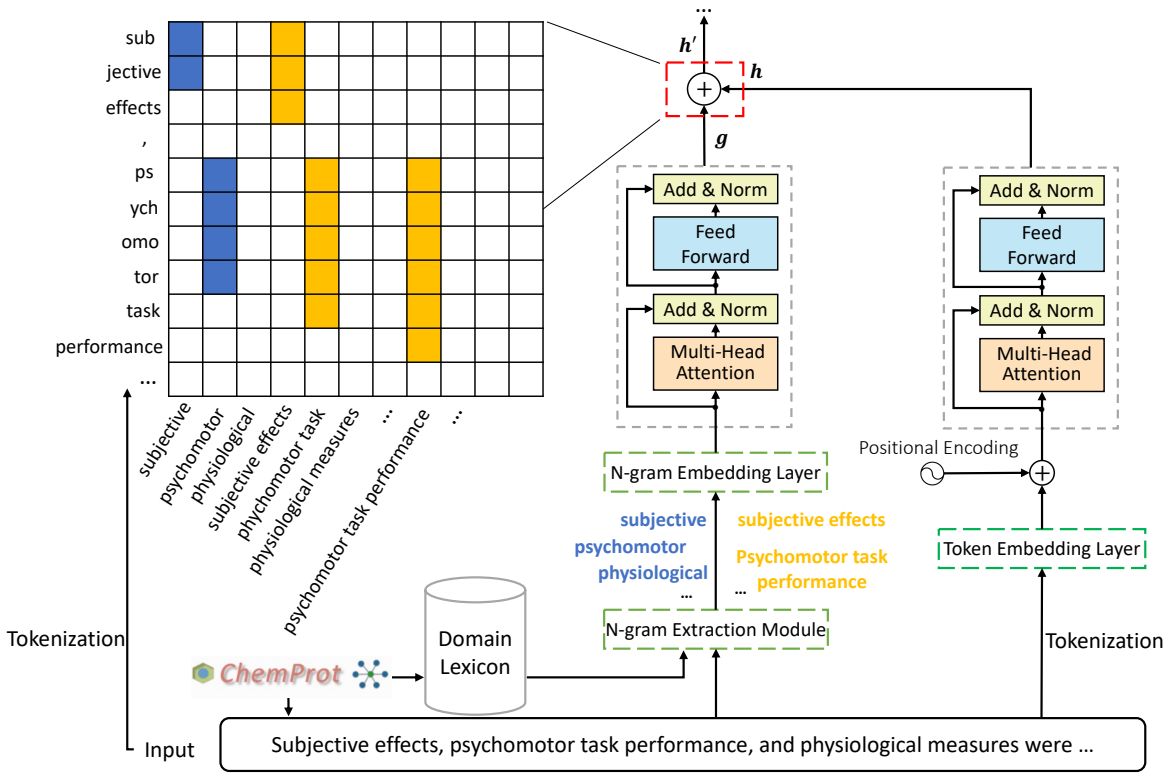

Figure 2: The overall architecture of our model.

as BERT or RoBERTa, provides a representation at the subword-level without any target domain knowledge. The right-hand side shows the proposed T-DNA to enhance the backbone pre-trained encoder, where word based n-grams in X are extracted from a pre-constructed lexicon $\mathcal{L}$, and are represented through n-gram attention module. The left-hand side shows the n-gram matching matrix and the integrating process of domain-specific representation and generic encoding.

In this section, we start with a detailed description of lexicon construction, then introduce our n-gram encoding module and how to integrate n-gram encoding with the backbone model to get domain-aware representation, and end with an illustration of two training strategies.

3.1 Lexicon Construction and N-gram Extraction

To better represent and incorporate unseen and domain-specific n-grams, we first need to find and extract them. Here we propose to use an unsupervised method, pointwise mutual information (PMI), to find domain-specific words and phrases by collocations and associations between words.

Given a sentence $\mathcal{X} = x_1 x_2 \cdots x_K$ with K words, for any two adjacent words (e.g., $\bar{x}$, $\widetilde{x}$) within the sentence, their PMI is calculated by

$$PMI(\bar{x}, \widetilde{x}) = \log \frac{p(\bar{x}\widetilde{x})}{p(\bar{x})p(\widetilde{x})}, \qquad (1)$$

where $p(x)$ is the probability of an n-gram x. When a high PMI score is detected between the adjacent $\bar{x}$ and $\widetilde{x}$, it suggests they are good collocation pairs, because they have a high probability of co-occurrence and are more likely to form an n-gram. On the contrary, a delimiter is inserted between the two adjacent words if their $PMI(\bar{x}, \widetilde{x})$ is less than a threshold σ, i.e., $\mathcal{X} = x_1 x_2 \cdots \bar{x}/\widetilde{x} \cdots x_K$. As a result, those consecutive words without a delimiter are identified as candidate domain-specific n-grams. After using PMI to segment each sentence in the training set of a target task, we could select among candidate n-grams to obtain the final n-gram lexicon $\mathcal{L}$, where each n-gram appears with a frequency of at least f.

In light of this lexicon, for each training input sentence $\mathcal{X} = t_1 t_2 \cdots t_i \cdots t_T$ with T tokens, where t_i denotes the i-th token of $\mathcal{X}$, we extract those sub-strings of $\mathcal{X}$ that exist in the lexicon to form domain-specific n-gram sequence $S = s_1 s_2, \cdots, s_j, \cdots, s_N$, with s_j indicating the j-th n-gram of $\mathcal{X}$. At the same time, an n-gram matching matrix, $\mathcal{M} \in R^{T \times N}$, can be built to record the

positions of the extracted domain-specific n-gram set and its associated tokens, where $m_{ij} = 1$ for $t_i \in s_j$ and $m_{ij} = 0$ for $t_i \notin s_j$. The matching matrix is shown in the left hand size of Figure 2.

3.2 Domain-aware Representation

The backbone pre-trained encoder is a Transformer architecture (Vaswani et al., 2017) with L layers, S self-attention heads and H hidden dimensions initialized from any pre-trained encoder (e.g., BERT or RoBERTa). The input sentence is passed through it, resulting in a generic hidden state h_i for each input token x_i. To get the domain-aware hidden representation, the n-gram adaptor network is implemented by a Transformer encoder with l layers, S self-attention heads and H hidden dimensions. First, the embeddings of domain-specific n-grams could be obtained by an n-gram embedding layer and then they are fed into the n-gram encoder to get a sequence of hidden states g via a multi-head attention mechanism. The n-gram encoder is able to model the interactions among all extracted n-grams and dynamically weighs n-grams to emphasize truly useful n-grams and ignores noisy information. The combination of the generic representation and domain-specific n-gram representation are computed by

$$h'_i = h_i + \sum_k g_{i,k}, \tag{2}$$

where h'_i is the desired domain-aware representation, and $g_{i,k}$ is the resulting hidden state for the i-th token and the k-th n-gram associated with this token according to the matching matrix $\mathcal{M}$. The n-gram encoding process and hidden state integration is repeated layer-by-layer along with the generic encoder for l layers from the bottom.

3.3 Training Strategies

Several training strategies could be used and we adopt two in our experiments: fine-tuning (FT) and task-adaptive pre-training (TAPT). For fine-tuning, we operate on the hidden state of the special classification token [CLS]. Following the tradition citation, we simply add a fully-connected layer as a classifier on top of the model and obtain the probabilities via a softmax layer. The classifier and the whole model are fine-tuned on the labeled task data in the target domain with cross-entropy loss. To inject unsupervised target domain knowledge, we leverage the task-adaptive pre-training proposed in (Gururangan et al., 2020) which strips the labels in downstream task training data and trains the model on this unlabeled data. We use the masked language model (MLM) as our objective and do not include the next sentence prediction (NSP) task following Liu et al. (2019); Lan et al. (2020).

Note that, our model also supports other training strategies such as domain-adaptive pre-training, which proves to be effective in Gururangan et al. (2020). One can pre-train our model on a far larger domain corpus (normally beyond 10GB) at the beginning, and then do the task-adaptive pre-training and fine-tuning. Because our main goal is to adapt our model in a low-resource setting in terms of data size and time cost, we leave it for future research.[7]

4 Experiment Settings

In this section, we first introduce eight benchmarking datasets. Then the baseline models, evaluation metrics, and implementation details are presented in the following three subsections, respectively.

4.1 Datasets

Following Gururangan et al. (2020), we conduct our experiments on eight classification tasks from four domains including biomedical sciences, computer science, news and reviews. The datasets are described as follows.

- **CHEMPROT** (Kringelum et al., 2016), a manually annotated chemical–protein interaction dataset extracted from 5,031 abstracts for relation classification.
- **RCT** (Dernoncourt and Lee, 2017), which contains approximately 200,000 abstracts from public medicine with the role of each sentence clearly identified.
- **CITATIONINTENT** (Jurgens et al., 2018), which contains around 2,000 citations annotated for their function.
- **SCIERC** (Luan et al., 2018), which consists of 500 scientific abstracts annotated for relation classification.
- **HYPERPARTISAN** (Kiesel et al., 2019), which contains 645 articles from Hyperpartisan news with either extreme left-wing or right-wing standpoint used for partisanship classification.
- **AGNEWS** (Zhang et al., 2015), consisting of 127,600 categorized articles from more than 2000 news source for topic classification.

[7] We show some analyses and discussion of data size in Section 6.2.

Domain		BioMed		CS		News		Reviews	
Dataset		CP	RCT	CI	SE	HP	AG	AM	IMDB
Train	S#	4.1K	1.8K	1.6K	3.2K	516	1.1K	1.1K	2.0K
	T#	895K	267K	376K	619K	1.7M	213K	1.0M	2.6M
	O.S#	4.1K	180K	1.6K	3.2K	516	115K	115K	20K
	O.T#	895K	27.4M	376K	619K	1.7M	21.4M	98.9M	25.9M
Dev	S#	2.4K	30K	114	455	64	5K	5K	5K
	T#	547K	4.6M	24K	89K	194K	929K	4.4M	6.6M
Test	S#	3.4K	30K	139	974	65	7.6K	25K	25K
	T#	773K	4.6M	31K	187K	238K	1.4M	21.5M	31.8M
Classes		13	5	6	7	2	4	2	2

Table 1: The statistics of the eight task datasets in four target domains. To limit the computational resources and maintain all datasets on thousand-level, we only take 10% of IMDB training set, and 1% of RCT, AG and AM training sets. O.S# and O.T# refer to the number of sentences and the number of tokens in the original datasets, respectively. S# denotes the number of sentences and T# is the number of tokens. CP, CI, SE, HP, AG and AM denote ChemProt, CitationIntent, SciERC, HyperPartisan,AGNews and Amazon, respectively.

- **Amazon** (McAuley et al., 2015), consisting of 145,251 reviews on Women's and Men's Clothing & Accessories, each representing users' implicit feedback on items with a binary label signifying whether the majority of customers found the review helpful.
- **IMDB** (Maas et al., 2011), 50,000 balanced positive and negative reviews from the Internet Movie Database for sentiment classification.

To create a low-resource setting, we constrain the size of all datasets into thousand-level. To do so, we randomly select a subset for RCT, AG, Amazon, IMDB with the ratio 1%, 1%, 1%, 10%, respectively. The details can be found in Table 1.

4.2 Baselines

In our experiments, the following two models serve as the main baselines.

- **RoBERTa+FT**: fine-tuned off-the-shelf RoBERTa-base model for downstream tasks.
- **RoBERTa+TAPT**: task-adaptive pre-trained on unlabeled task data starting from RoBERTa and then fine-tuned on labeled data.

4.3 Evaluation Metrics

Following Beltagy et al. (2019), we adopt macro-F1 for CitationIntent, SciERC, HyperPartisan, AGNews, Amazon, IMDB, and micro-F1 for ChemProt and RCT as evaluation metrics. Macro-F1 will compute the F1 metric independently for each class and then take the average, whereas micro-F1 will aggregate the contributions of all classes to compute the average metric. In a multi-class classification setup, micro-F1 is preferable if there is class imbalance, which is true for ChemProt and RCT.

4.4 Implementation

We implement the RoBERTa-base architecture and initialize it with pre-trained weights by Huggingface's Transformers library[8]. In order to obtain a fast and warm start for n-gram representations, we utilize fastText (Bojanowski et al., 2017) to initialize n-gram embeddings. Considering the small amount of data and based on our experience, the number of N-gram encoding layers l is set to 1.

For unsupervised task-adaptive pre-training (TAPT), the batch size is set to 16 and training epochs range from 10 to 15. We adopt Adam (Kingma and Ba, 2015) as the optimizer , where the corresponding learning rates of different datasets can be found in our code. The dropout rate is set to 0.5. For the task-specific fine-tuning (FT), we use similar hyperparameter settings and the details are elaborated in the Appendix. All the experiments are implemented on Nvidia V100 GPUs.

5 Experimental Results

We compare the performance of the RoBERTa model with and without T-DNA on the aforementioned datasets. In both fine-tuning and task adaptive pre-training experiments, T-DNA shows significant improvements over the pre-trained generic RoBERTa.

[8]https://github.com/huggingface/transformers

DOMAIN	BIOMED		CS		NEWS		REVIEWS	
DATASET	CP	RCT	CI	SE	HP	AG	AM	IMDB
RoBERTa+FT	81.10$_{0.70}$	80.72$_{0.40}$	56.74$_{5.47}$	74.06$_{5.25}$	88.15$_{1.51}$	88.60$_{0.01}$	63.04$_{0.69}$	92.29$_{0.23}$
+T-DNA	**82.66**$_{0.31}$	**81.52**$_{0.41}$	**64.95**$_{4.98}$	**78.61**$_{2.00}$	**92.49**$_{0.69}$	**88.91**$_{0.06}$	**63.92**$_{0.62}$	**92.91**$_{0.71}$
RoBERTa+TAPT	82.24$_{1.33}$	82.73$_{0.23}$	63.44$_{2.30}$	77.85$_{1.12}$	92.70$_{0.73}$	88.84$_{0.01}$	64.13$_{0.22}$	92.77$_{0.25}$
+T-DNA	**83.89**$_{0.76}$	**83.94**$_{0.27}$	**69.73**$_{2.87}$	**79.40**$_{0.48}$	**93.91**$_{1.48}$	**89.05**$_{0.03}$	**64.36**$_{0.34}$	**93.13**$_{0.15}$

Table 2: The overall performance of T-DNA and the comparison against existing models on eight target downstream datasts. We report average scores across five random seeds, with standard deviations as subscripts.

5.1 Fine-Tuning

The results of fine-tuning on eight datasets are reported in Table 4. In general, the RoBERTa model with T-DNA outperforms that without T-DNA on all datasets, clearly indicating the effectiveness of T-DNA by emphasizing multi-granularity information. On average, T-DNA is able to bring an improvement of performance by around 2.66%.

Across all eight datasets, it is observed that T-DNA achieves the greatest improvement (8.21%) on the CitationIntent dataset and the least improvement on the AGNews dataset. One reasonable explanation for different improvements is that the domain gap between the RoBERTa pre-training domain and the CS domain is the greatest so that far more gains could be obtained by an effective adaptation strategy. To confirm this, we follow Gururangan et al. (2020) to characterize the domain similarity by analyzing vocabulary overlap and we draw the same conclustion that RoBERTa's pre-training domain has a similar vocabulary to News and Reviews, but far more dissimilar vocabulary to BioMed and CS. In light of this observation, we recognize that the proposed method is more applicable when the domain gap is large. In this scenario, the potential of incorporating multi-grained information by domain-specific n-grams is greatly exploited to boost the performance of adaptation.

When comparing the improvements over four domains, T-DNA is able to offer 1.18%, 6.38%, 2.33%, 0.75% gains on BioMed, CS, News, Reviews, respectively. The improvement on the CS domain is the best while on the Reviews domain it is the poorest, which is consistent with previous analyses across datasets for similar reasons.

5.2 Task-Adaptive Pre-Training

In the previous section, we show that T-DNA is helpful in fine-tuning. Additionally, we would like to explore whether T-DNA is complementary to more training strategies, such as task-adaptive pre-training (TAPT). TAPT has been shown useful for

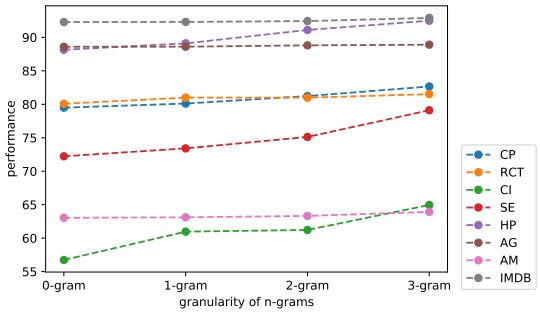

Figure 3: Effects of Different Granularities (N=0,1,2,3).

pre-trained models in previous studies (Howard and Ruder, 2018; Gururangan et al., 2020), by pre-training on the unlabeled task dataset drawn from the task distribution. The experimental results of two models with and without T-DNA are reported in the bottom two rows in Table 4. From the results, we can clearly see that the model with T-DNA achieves better performance on all datasets compared to the generic RoBERTa model without T-DNA. The T-DNA helps to improve the performance by approximately 1.59% on average, which shows that the effectiveness of T-DNA does not vanish when combined with TAPT. Instead, it further leads to a large performance boost for pre-trained models, indicating that T-DNA is a complementary approach, where explicitly modeling domain-specific information helps the unsupervised learning of representations (i.e., the masked language model (MLM) pre-training objective).

Overall, for both FT and TAPT experiments, the results show that T-DNA significantly improves domain adaptation performance based on a generic pre-trained model. We attribute this improvement to the essential domain-specific semantic information that is carried by n-grams and the valid representation of n-grams from the T-DNA network.

6 Analyses

We analyze several aspects of T-DNA, including the effects of different granularities and the effects

Task	RCT		AG		AM		IMDB	
Model	w.o	w.	w.o	w.	w.o	w.	w.o	w.
10%	80.78	$82.23^{\uparrow 1.45}$	90.11	$92.01^{\uparrow 1.90}$	63.13	$64.10^{\uparrow 0.97}$	92.29	$92.91^{\uparrow 0.62}$
20%	85.22	$86.16^{\uparrow 0.94}$	91.71	$92.14^{\uparrow 0.43}$	64.01	$65.12^{\uparrow 1.11}$	92.11	$92.89^{\uparrow 0.78}$
50%	87.10	$87.69^{\uparrow 0.59}$	92.17	$92.58^{\uparrow 0.41}$	65.52	$66.10^{\uparrow 0.58}$	93.13	$93.32^{\uparrow 0.19}$
100%	87.31	$87.69^{\uparrow 0.38}$	93.75	$94.00^{\uparrow 0.25}$	66.79	$67.14^{\uparrow 0.35}$	94.34	$94.81^{\uparrow 0.47}$

Table 3: Performance gains of T-DNA w.r.t. different sampling ratios of RCT, AG, AM and IMDB datasets. w. and w.o indicate whether the model is equipped with T-DNA or not. The uparrow marks where a positive gain is obtained.

of data size. In addition, we examine the attention mechanism to verify the effects of n-gram representations during the domain shift. The details are illustrated in this section.

6.1 Effects of Different Granularities

The lexical unit in RoBERTa is a subword obtained from byte pair encoding (BPE) (Sennrich et al., 2016) tokenization, resulting in a smaller token space and more training data for each token. Our approach provides coarse-grained information carried by the larger lexical units, n-gram.

To verify the contribution of larger granularity information, we compare the improvement brought by T-DNA with information of different granularities, for n from 0 to 3. Note that here n means that we extract and incorporate all n-grams with a length smaller or equal to n (within a certain granularity). For example, $n = 3$ means that we include all unigrams, bigrams and trigrams. Two consistent observations could be made. First, adding only 1-gram is able to bring improvements over 0-gram (i.e., without T-DNA) on all eight datasets, as shown in Figure 3. As we know, the tokens in the generic encoder are at the subword-level and our unigrams are at the word-level, which can be seen as a combination of subwords. Therefore, the results suggest that adding unseen words through our adaptor network is effective, which could enhance the interaction between subwords of the same word, especially for the new words in the target domain.

Moreover, based on 1-gram, involving larger granularity offer further gains. Comparing 2-gram and 3-gram v.s. 1-gram, the consistent improvements of T-DNA demonstrate that the potential boundary information presented by n-grams plays an essential role in learning representations by providing explicit and better guidance.

6.2 Effects of Data Size

In the previous section, we explored the virtue of incorporating multi-grained information under resource-limited settings, where only a small subset of specific datasets can be accessed. In addition, we are curious whether T-DNA could work well on a larger scale. To this end, we sample different ratios (i.e., 10%, 20%, 50%, 100%) of four datasets (i.e., RCT, AGNews, Amazon and IMDB) and investigate how T-DNA performs at different data scales. As shown in Table 3, the model with T-DNA always outperforms that without T-DNA w.r.t. any subsets of four datasets. This demonstrates that models with T-DNA could easily adapt to any size of dataset with the help of domain-specific n-gram information. However, it is also noted that the performance gains of our method decayed with the increase of the amount of training data, dropping from 1.24% (proportion=10%) to 0.36% (proportion=100%). It is not surprising because with adequate data, a model is able to learn a good representation with supervised learning without the need of prior knowledge. However, since sufficient data normally could not be accessed in reality, especially labeled data, we argue that T-DNA is desirable and necessary for domain adaptation.

6.3 Visualization of N-gram Representations

To verify the effects of n-gram representations during the domain shift, we examine the attention mechanism of RoBERTa and T-DNA by plotting the attention maps and salience maps using the LIT tool (Tenney et al., 2020). In the attention map of RoBERTa without T-DNA, we found that the tokens usually improperly attend to other tokens in the sentence. For example, in Figure 4, "*Barbie*" attributes more attentions to "*animated*" and "*scary*" but omits "*creepy*" and fails to capture "*scary as hell*" as an integrated phase. In contrast, when the model is equipped with T-DNA, this variant will shift its attention to include "*creepy*" and

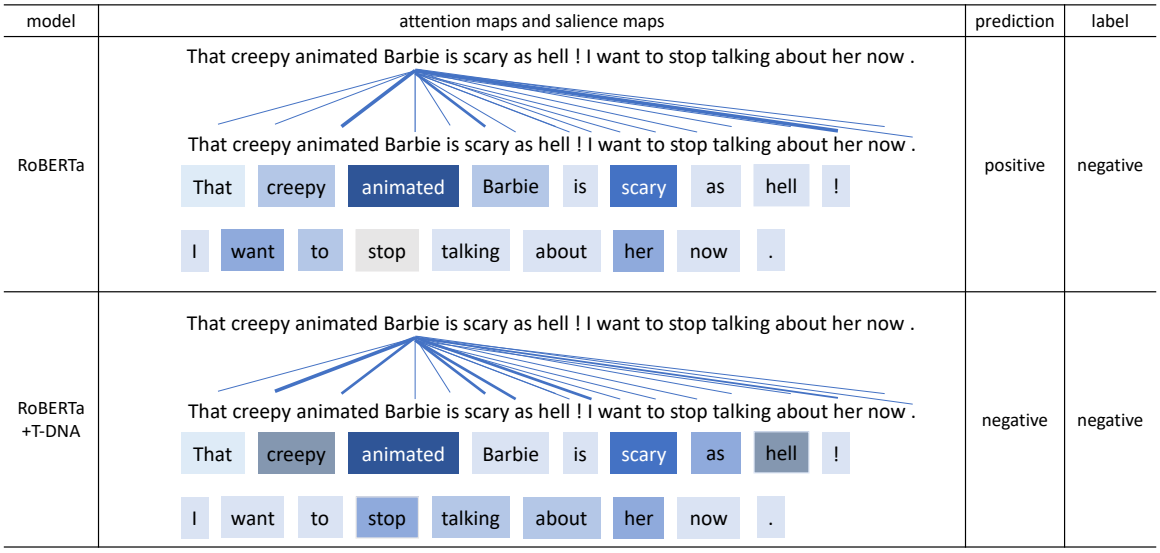

Figure 4: The visualization of attention maps and salience maps of RoBERTa and RoBERTa+T-DNA. The upper region of each row shows the attention map, where thicker lines denote higher attention weights. The bottom region illustrates the salience map, where the darker color box denotes the more dominant weights for the prediction.

force the model to focus on the informative phrase "*scary as hell*". Furthermore, the salience map of RoBERTa without T-DNA suggests that "*animated*" and "*scary*" dominate its prediction while "*creepy*" and "*scary as hell*" are captured by our T-DNA, which is consistent with the decision process of human beings.

Due to the space limitations, more visualized examples are not shown here. However, based on considerable empirical evidence, we conclude that the unreliable representations of domain-specific n-grams (words and phrases) might be one of the main causes for model degradation.

7 Related Work

A large performance drop of pre-trained models caused by domain shift has been observed and many domain-specific BERT models (Beltagy et al., 2019; Alsentzer et al., 2019; Huang et al., 2019; Lee et al., 2020) have been introduced to bridge the domain gap. For example, SciBERT (Beltagy et al., 2019) is trained on 1.14M scientific papers from Semantic Scholar corpus (Ammar et al., 2018) for 7 days on TPU v3-8 machine and BioBERT (Lee et al., 2020) is trained on PubMed abstracts and PMC full text articles for 23 days on eight NVIDIA V100 GPUs. ClinicalBERT (Alsentzer et al., 2019) is trained on about 2 million notes in the MIMIC-III v1.4 database (Johnson et al., 2016) for 17-18 days on a single GeForce GTX TITAN X 12 GB GPU. However, they all incur a huge computational cost, which is not affordable for many university labs or institutions. This is precisely why we believe that our efficient adaptor is useful to the community. Although Gururangan et al. (2020) introduced task-adaptive pre-training (TAPT) to save time by training on unlabeled downstream task data, we demonstrate that our plug-in adaptor is faster and more effective because of the explicit learning strategy and efficient model architecture.

Out of vocabulary (OOV) words refer to those words that are not in the vocabulary list and have received a lot of attention in recent years. One way to handle OOV words is to simply utilize and learn an "unknown" embedding during training. Another way is to add in-domain words into the original vocabulary list and learn their representation by pre-training from scratch (Beltagy et al., 2019; Gu et al., 2020), which requires substantial resources and training data. Moreover, SciBERT (Beltagy et al., 2019) found that in-domain vocabulary is helpful but not significant while we attribute it to the inefficiency of implicit learning of in-domain vocabulary. To represent OOV words in multilingual settings, the mixture mapping method (Wang et al., 2019) utilized a mixture of English subwords embedding, but it has been shown useless for domain-specific

words by Tai et al. (2020). ExBERT (Tai et al., 2020) applied an extension module to adapt an augmenting embedding for the in-domain vocabulary but it still needs large continuous pre-training. Similar to our work, they highlight the importance of the domain-specific words but all of these work neither explore the understanding of performance drop during a domain shift nor examine the importance of multi-grained information. Large granularity contextual information carried by spans or n-grams has proven to be helpful to enhance text representation for Chinese (Song et al., 2009; Song and Xia, 2012; Ouyang et al., 2017; Kim et al., 2018; Peng et al., 2018; Higashiyama et al., 2019; Tian et al., 2020e,b; Li et al., 2020; Diao et al., 2020; Song et al., 2021) and English (Joshi et al., 2020; Xiao et al., 2020; Tian et al., 2020c,d). In addition to text encoders on pre-training, the kNN-LM (Khandelwal et al., 2019) proposes to augment the language model for effective domain adaptation, by varying the nearest neighbor datastore of similar contexts without further training. However, all of the previous studies focused on either general pre-training procedures or different tasks (e.g., language modeling), and did not explore the effectiveness of multi-grained information for domain adaptation. We hence view them as orthogonal to our work.

8 Conclusion

In this work, we first reveal a novel discovery behind the performance drop during a domain shift, demonstrating that an unreliable representation of domain-specific n-grams causes the failure of adaptation. To this end, we propose an innovative adaptor network for generic pre-trained encoders, supporting many training strategies such as task-adaptive pre-training and fine-tuning, both leading to significant improvements to eight classification datasets from four domains (biomedical, computer science, news and reviews). Our method is easy to implement, simple but effective, implying that explicitly representing and incorporating domain-specific n-grams offer large gains. In addition, further analyses consistently demonstrate the importance and effectiveness of both unseen words and the information carried by coarse-grained n-grams.

Acknowledgments

This work was supported by the General Research Fund (GRF) of Hong Kong (No. 16201320). The authors also want to thank the Sinovation Ventures for their great support. Y. Song was supported by NSFC under the project "The Essential Algorithms and Technologies for Standardized Analytics of Clinical Texts" (12026610) and Shenzhen Institute of Artificial Intelligence and Robotics for Society under the project "Automatic Knowledge Enhanced Natural Language Understanding and Its Applications" (AC01202101001). R. Xu was supported by the Hong Kong PhD Fellowship Scheme (HKPFS).

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. 2018. Construction of the Literature Graph in Semantic Scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 84–91.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. 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 3606–3611.

Piotr Bojanowski, Édouard Grave, Armand Joulin, and Tomáš Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Franck Dernoncourt and Ji Young Lee. 2017. PubMed 200k RCT: a Dataset for Sequential Sentence Classification in Medical Abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 308–313.

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.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2020. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. In *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing: Findings, pages 4729–4740.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *arXiv e-prints*, pages arXiv–2007.

Suchin Gururangan, Ana Marasović, 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Shohei Higashiyama, Masao Utiyama, Eiichiro Sumita, Masao Ideuchi, Yoshiaki Oida, Yohei Sakamoto, and Isaac Okada. 2019. Incorporating Word Attention into Character-Based Word Segmentation. 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 2699–2709, Minneapolis, Minnesota.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. *arXiv preprint arXiv:1904.05342*.

AE Johnson, TJ Pollard, L Shen, LW Lehman, M Feng, M Ghassemi, B Moody, P Szolovits, LA Celi, and RG Mark. 2016. MIMIC-III, a Freely Accessible Critical Care Database. *Scientific data*, 3:160035–160035.

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.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the Evolution of a Scientific Field through Citation Frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations*.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 Task 4: Hyperpartisan News Detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839.

Geewook Kim, Kazuki Fukui, and Hidetoshi Shimodaira. 2018. Word-like Character N-gram Embedding. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 148–152.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

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.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. FLAT: Chinese NER using Flat-Lattice Transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842, Online.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts.

2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based Recommendations on Styles and Substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52.

En Ouyang, Yuxi Li, Ling Jin, Zuofeng Li, and Xiaoyan Zhang. 2017. Exploring N-gram Character Presentation in Bidirectional RNN-CRF for Chinese Clinical Named Entity Recognition. In *CEUR Workshop Proc*, volume 1976, pages 37–42.

Haiyun Peng, Yukun Ma, Yang Li, and Erik Cambria. 2018. Learning Multi-grained Aspect Target Sequence for Chinese Sentiment Analysis. *Knowledge-Based Systems*, 148:167–176.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Yan Song, Chunyu Kit, and Xiao Chen. 2009. Transliteration of Name Entity via Improved Statistical Translation on Character Sequences. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 57–60, Suntec, Singapore.

Yan Song and Fei Xia. 2012. Using a Goodness Measurement for Domain Adaptation: A Case Study on Chinese Word Segmentation. In *LREC*, pages 3853–3860.

Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. 2021. ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders. *arXiv preprint arXiv:2105.01279*.

Wen Tai, HT Kung, Xin Luna Dong, Marcus Comiter, and Chang-Fu Kuo. 2020. exBERT: Extending Pretrained Models with Domain-specific Vocabulary Under Constrained Training Resources. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1433–1439.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, et al. 2020. The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models. *arXiv preprint arXiv:2008.05122*.

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 2020a. Joint Chinese Word Segmentation and Part-of-Speech Tagging via Two-way Attentions of Auto-analyzed Knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8286–8296.

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 2020b. Joint Chinese Word Segmentation and Part-of-speech Tagging via Two-way Attentions of Auto-analyzed Knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8286–8296, Online.

Yuanhe Tian, Yan Song, and Fei Xia. 2020c. Supertagging combinatory categorial grammar with attentive graph convolutional networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6037–6044.

Yuanhe Tian, Yan Song, Fei Xia, and Tong Zhang. 2020d. Improving Constituency Parsing with Span Attention. In *Findings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020e. Improving Chinese Word Segmentation with Wordhood Memory Networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8274–8285, Online.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in neural information processing systems*, pages 5998–6008.

Hai Wang, Dian Yu, Kai Sun, Jianshu Chen, and Dong Yu. 2019. Improving Pre-Trained Multilingual Model with Vocabulary Expansion. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 316–327.

Dongling Xiao, Yu-Kun Li, Han Zhang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE-Gram: Pre-Training with Explicitly N-Gram Masked Language Modeling for Natural Language Understanding. *arXiv preprint arXiv:2010.12148*.

Ze Yang, Wei Wu, Can Xu, Xinnian Liang, Jiaqi Bai, Liran Wang, Wei Wang, and Zhoujun Li. 2020. StyleDGPT: Stylized Response Generation with Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1548–1559.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. *Advances in neural information processing systems*, 28:649–657.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. DIALOGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.

A Description of Computing Infrastructure

All the experiments are implemented on Nvidia V100 GPUs with 32GB memory.

B Run Time

DOMAIN	BIOMED		CS		NEWS		REVIEWS	
DATASET	CP	RCT	CI	SE	HP	AG	AM	IMDB
RoBERTa+FT	95	40	37	74	50	102	130	114
+T-DNA	93	39	40	72	52	104	131	113
RoBERTa+TAPT	300	132	117	234	285	389	402	392
+T-DNA	320	128	114	240	290	390	400	394

Table 4: Running time per epoch of models, in the unit of second.

C Validation Performance

DOMAIN	BIOMED		CS		NEWS		REVIEWS	
DATASET	CP	RCT	CI	SE	HP	AG	AM	IMDB
RoBERTa+FT	80.08	81.21	58.06	75.33	93.50	88.70	62.50	93.04
+T-DNA	81.17	82.00	62.98	79.62	91.81	88.64	63.40	92.83
RoBERTa+TAPT	81.27	80.98	60.11	77.08	93.50	88.90	64.30	92.38
+T-DNA	82.58	83.24	67.89	80.69	93.74	89.31	64.27	93.11

Table 5: The validation performance.

D Evaluation Measures

We use manual tuning and adopt macro-F1 for CitationIntent, SciERC, HyperPartisan, AGNews, Amazon, IMDB, and micro-F1 for ChemProt and RCT as evaluation metrics. Macro-F1 will compute the F1 metric independently for each class and then take the average, whereas micro-F1 will aggregate the contributions of all classes to compute the average metric. In a multi-class classification setup, micro-F1 is preferable if there is class imbalance, which is true for ChemProt and RCT.

E Bounds of Hyperparameters

Hyperparameter	**Assaignment**
number of epochs	3(FT) or 15(TAPT)
patience	1
batch size	[4,8,16,32,64]
learning rate	[1e-5,1e-4]
dropout	0.5
classification layer	[1,2]
learning rate optimizer	Adam
Adam epsilon	1e-8
Adam beta	0.9, 0.999
learning rate optimizer	Adam

Table 6: Bounds of hyperparameters.

F Configuration of Best Model

Hyperparameter	Assaignment
number of epochs	3(FT) or 15(TAPT)
patience	1
batch size	32
learning rate	4e-5
dropout	0.5
classification layer	1
learning rate optimizer	Adam
Adam epsilon	1e-8
Adam beta	0.9, 0.999
learning rate optimizer	Adam

Table 7: Configuration of the best model.