Multidomain Pretrained Language Models for Green NLP

Antonis Maronikolakis
CIS, LMU Munich
antmarakis@cis.lmu.de

Hinrich Schütze
CIS, LMU Munich

Abstract

When tackling a task in a given domain, it has been shown that adapting a model to the domain using raw text data before training on the supervised task improves performance versus solely training on the task. The downside is that a lot of domain data is required and if we want to tackle tasks in n domains, we require n models each adapted on domain data before task learning. Storing and using these models separately can be prohibitive for low-end devices. In this paper we show that domain adaptation can be generalised to cover multiple domains. Specifically, a single model can be trained across various domains at the same time with minimal drop in performance, even when we use less data and resources. Thus, instead of training multiple models, we can train a single multidomain model saving on computational resources and training time.

1 Introduction

Domain adaptation in the form of training on unlabeled data has been prevalent in recent work (Rietzler et al., 2020; Han and Eisenstein, 2019). When given a task T in domain D, it is useful to adapt our model on raw text data pertinent to D before supervised training on the labeled data of T.

Unfortunately, domain adaptation is expensive and costly. So even though results do get better, there needs to be a balance between use of computational resources and model performance.

Accentuating the issue is the rising dominance of increasingly larger models (Devlin et al., 2019; Brown et al., 2020). These large models require a lot of data and computational resources to train. Not only have these resources been prohibitive for smaller labs, but the environmental impact of training such large models cannot be understated either (Strubell et al., 2019; Lacoste et al., 2019).

There is therefore a need to build models that can tackle tasks across multiple domains, much in the same way multilingual models are able to operate across multiple languages. These multidomain models, to be useful, need to exhibit performance comparable to models adapted to a single domain. Then, these models can be trained and deployed with reduced computational costs.

In this paper we explore such multidomain models. We compare multidomain DistilBERT (Sanh et al., 2019) models with single-domain DistilBERT models[1]. In our analysis, we test the multidomain model on tasks from multiple domains (including MultiNLI), we examine how important adaptation order is for performance and we show that training on the domains jointly or sequentially does not impact effectiveness. We also reproduce findings in Rietzler et al. (2020), where the authors showed that pretraining on an irrelevant domain is not beneficial. Finally, we show that much like multilingual BERT, multidomain models can be used for better performance in low-resource domains, with the low-resource domain leveraging the larger pretrained model.

2 Related Work

It has been shown empirically that domain adaptation improves model performance (Gururangan et al., 2020). Specifically, the authors experimented on RoBERTa (Liu et al., 2019) and eight tasks ranging across four domains (computer science and biomedical papers, reviews and news). Performance increased when adapting to a domain pertinent to the domain of the task, compared to when the two domains were not as related.

Work has also been done to alleviate the environmental impact of training larger models. In Poerner et al. (2020), an inexpensive method was proposed for domain adaptation, which can be performed on

[1] Code and instructions for data acquisition are available at https://github.com/antmarakis/multidomain_green_nlp.

Proceedings of the Second Workshop on Domain Adaptation for NLP, pages 1–8
April 20, 2021. ©2021 Association for Computational Linguistics

a CPU and significantly reduces training cost.

Finally, MobileBERT is a downsized BERT model with around 4 times fewer parameters (Sun et al., 2020) than the original BERT-base model. There is a need to fit larger models in low-capacity machines, with MobileBERT and our work being a step in that direction, alongside other work in the pruning area (Sanh et al., 2020; Zhao et al., 2020).

3 Data

3.1 Raw Text Domain Data

In Gururangan et al. (2020), the authors showed that adaptation using domain data improves performance on a downstream task. In our work, we trained two types of models, *single-domain* models (one for each domain) and a *multidomain* model. All models were pretrained from scratch.

For the single-domain models, we used around 4GBs of each dataset, adapting DistilBERT on each domain for 1 epoch. This resulted in four distinct models. For our multidomain model, we adapted our model successively on all domains, using around half of the available data (approximately 2GBs from each domain) for 1 epoch as well.

The datasets we used for adaptation are: *Amazon* (He and McAuley, 2016), *Arxiv* (Cohan et al., 2018), *Realnews* (Zellers et al., 2019), *CS* (Lo et al., 2020) and *Reddit Comments* (Völske et al., 2017).

The original datasets were truncated for our experiments, using approximately only the first 4GBs of text. For the *Amazon* dataset, which contains reviews across multiple products (eg. books, music, clothing), we took care to balance data across all categories, by sampling at random n=50,000 reviews from products with more reviews than n.

3.2 Supervised Task Data

Models were evaluated on eight supervised classification tasks in total, spanning four domains. An overview and description of the tasks can be found in Appendix A. Each model was trained on each task separately. For most of the datasets, train/dev/test splits are already provided. Where such splits are not available, we randomly sample 60/20/20 sets from the original data for train/dev/test splits respectively.

4 Training

The training of our models is broken up in two steps: domain adaptation and supervised task learning. Furthermore, we have two setups for our experiments: a) perform domain adaptation on four models in parallel and then train them on each individual task, and b) perform domain adaptation on a single model for all domains successively before task learning.

We also perform ablation studies on irrelevant and low-resource domain adaptation, domain adaptation order and finally investigate whether joint instead of sequential adaptation performs better.

4.1 Domain Adaptation

In our experiments we used DistilBERT (Sanh et al., 2019). DistilBERT is a lightweight BERT-based model that showcases performance remarkably close to BERT, with around 40% fewer parameters. Training was performed on a publicly available[2] pretrained model.

First, we perform domain adaptation on each of the four domains separately. This results in four distinct models, each adapted to a different domain. Hyperparameters can be found in Appendix B. We also train a single model successively on all four domains, resulting in a multidomain model. All possible domain adaptation orders were compared and for our comparisons we chose the order with the worst overall performance, which is *Amazon* → *Reddit Comments* → *Realnews* → *Arxiv* (*Am-RC-R-Ar*). Further details on order comparison can be found in 5.4.

4.2 Supervised Task Learning

After the domain adaptation phase, the single-domain and multidomain models were trained on the training data of each task. Training took place over 1 or 2 epochs, trying to keep training time approximately equal across all tasks. Namely, all tasks required 1 epoch to train, except the *ACL-ARC* and *HyperPartisan* tasks, for which the models were trained for 2 epochs. More details on the hyperparameters are available in Appendix C.

5 Results

5.1 Base vs. Single-domain vs. Multidomain

Evaluation took place across eight tasks, covering all four domains. As an added experiment, we also evaluated on the MultiNLI dataset (Williams et al., 2018). MultiNLI is a dataset for textual entailment consisting of sentence pairs spanning multiple genres. We made three runs over each task and averaged the results, shown in Table 1.

[2] Accessible online here.

Note that the multidomain model used here (*Am-RC-R-Ar*) comes from the lowest-performing domain adaptation order. We chose it to illustrate that even in the worst case scenario, there is still a substantial improvement over the base model. On average, the gains are even larger (Section 5.4).

	Base	**Single**	**Multi**
ACL-ARC	$68.0_{0.3}$	$\mathbf{74.1_{1.5}}$	$70.5_{2.1}$
SciCite	$84.9_{0.8}$	$86.0_{0.3}$	$\mathbf{86.1_{0.2}}$
SARC	$75.3_{0.1}$	$\mathbf{78.6_{0.6}}$	$76.1_{0.3}$
TalkDown	$86.1_{0.7}$	$\mathbf{86.5_{0.1}}$	$86.1_{0.7}$
HyperPartisan	$78.9_{0.8}$	$\mathbf{81.4_{2.0}}$	$80.5_{1.4}$
AG-News	$94.0_{0.1}$	$94.1_{0.1}$	$\mathbf{94.3_{0.1}}$
IMDB	$86.3_{0.1}$	$\mathbf{87.2_{0.2}}$	$86.7_{0.1}$
Clothing	$68.9_{0.2}$	$\mathbf{69.8_{0.3}}$	$69.3_{0.3}$
MultiNLI	$77.0_{0.4}$	$77.8^{*}_{0.2}$	$\mathbf{79.1_{0.3}}$
Average	79.7	**81.7**	81.1

Table 1: Accuracy in percentage for task/model combinations. Standard deviation is shown in subscript (based on three runs). With *Base* we denote the original model, with *Single* the model trained on the corresponding domain and with *Multi* the multidomain model. For the *Single* model, we show the best accuracy out of all the models in MNLI.

When adapting DistilBERT to a single domain, performance is greater compared to the base model. When adapting to all domains, performance still increases, although by less on average. The average increase from base DistilBERT to single-domain DistilBERT is 2.0, whereas the multidomain model shows an improvement of 1.4 over the respective base model.

Overall, performance increases across tasks when adapting to all domains and in some cases the multidomain model is better than the single-domain model. Also, performance never drops for the multidomain model which, in the worst cases, still achieves marginally higher accuracy than the base model. At the same time, on MultiNLI the multidomain model scores higher than both base and all single-domain models, showcasing its domain-agnostic capabilities.

It is thus shown that multidomain models provide a boost in most tasks while never hindering performance in any task. They do so while requiring less data; from the 16GBs needed to train the four single models (4GBs for each domain), we only require 8GBs (2GBs for each domain) to train the multidomain model with comparable results.

5.2 Low-resource Domains

One of the advantages of multilingual models is that low-resource languages can leverage the multilingual model. A model is trained on multiple languages before the low-resource language is added. The resulting model performs better than a model trained only on the low-resource language.

We examine whether this holds true for multidomain models as well. For this experiment, we include the biomedical domain by further adapting our multidomain model to the *Pubmed* (Lo et al., 2020) dataset. Namely, we experiment with 10MB, 100MB and 500MB *Pubmed* datasets added to *Am-RC-R-Ar*. After pretraining on each of the new, smaller *Pubmed* sets, we test our models on *ChemProt* (Kringelum J, 2016) and *Pubmed-RCT* (Dernoncourt and Lee, 2017). As a baseline, we pretrain DistilBERT on solely the *Pubmed* datasets. We also evaluate how the original multidomain model does without any biomedical data. Finally, we examine if training on the low-resource domain has a catastrophic effect on the previously learned domains. Results are shown in Table 2.

Due to differences in dataset sizes, care was taken to keep training times approximately equal for all setups. For *Pub-500*, we trained for 1 epoch. For *Pub-100* we trained for 5 epochs and for *Pub-10* we trained for 50 epochs.

When pretraining only on *Pubmed*, the amount of training data used does not have an impact on performance. All of *Pub-10/100/500* perform similarly. Performance on *ChemProt* is higher than the multidomain model by around 0.6, while on the rest of the tasks accuracy is not as high.

If we further adapt the multidomain model to the *Pubmed* sets, we get a larger improvement over the original multidomain model, not only on the tasks in the biomedical domain, but over all examined tasks. In *ChemProt*, the improvement is around 2.4 over the original multidomain model and around 1.0 over the single-domain *Pubmed* models. For the rest of the tasks we see marginal improvements across the board and on average the new multidomain model (*Am-RC-R-Ar-P*) performs the best out of all models.

Results here indicate that performance is improved when continuously adapting to a low-resource domain than simply pretraining a model on it, regardless of domain data size. In fact, performance improves overall, possibly because of the increased amount of training data.

	Multi	**Pub-10**	**Pub-100**	**Pub-500**	**+Pub-10**	**+Pub-100**	**+Pub-500**
ACL-ARC	$70.5_{2.1}$	$68.7_{1.7}$	$68.5_{1.4}$	$68.5_{2.0}$	$70.7_{1.2}$	$70.7_{1.8}$	$\mathbf{71.0_{1.7}}$
SciCite	$86.1_{0.2}$	$85.7_{0.2}$	$85.9_{0.3}$	$85.5_{0.2}$	$86.1_{0.2}$	$\mathbf{86.3_{0.3}}$	$86.2_{0.2}$
SARC	$\mathbf{76.1_{0.3}}$	$75.2_{0.2}$	$75.2_{0.2}$	$75.7_{0.1}$	$\mathbf{76.1_{0.3}}$	$\mathbf{76.1_{0.3}}$	$\mathbf{76.1_{0.2}}$
TalkDown	$86.1_{0.7}$	$86.0_{0.5}$	$85.9_{0.4}$	$86.0_{0.5}$	$86.1_{0.3}$	$\mathbf{86.2_{0.4}}$	$86.1_{0.4}$
HyperPartisan	$80.5_{1.4}$	$78.8_{1.2}$	$79.0_{2.3}$	$78.8_{1.3}$	$80.5_{1.8}$	$80.5_{2.0}$	$\mathbf{80.6_{1.6}}$
AG-News	$\mathbf{94.3_{0.1}}$	$94.0_{0.1}$	$94.1_{0.1}$	$94.0_{0.1}$	$\mathbf{94.3_{0.1}}$	$\mathbf{94.3_{0.1}}$	$\mathbf{94.3_{0.1}}$
IMDB	$86.7_{0.1}$	$86.4_{0.1}$	$86.3_{0.1}$	$86.3_{0.1}$	$86.7_{0.1}$	$\mathbf{86.8_{0.1}}$	$86.7_{0.1}$
Clothing	$69.3_{0.3}$	$66.7_{0.4}$	$67.2_{0.5}$	$66.9_{0.3}$	$\mathbf{69.5_{0.4}}$	$69.3_{0.4}$	$69.4_{0.3}$
ChemProt	$77.0_{0.6}$	$78.3_{0.1}$	$78.5_{1.1}$	$78.3_{0.2}$	$79.2_{0.1}$	$79.4_{0.5}$	$\mathbf{79.6_{0.4}}$
Pubmed-RCT	$86.4_{0.1}$	$86.4_{0.1}$	$86.4_{0.1}$	$86.4_{0.1}$	$86.4_{0.0}$	$\mathbf{86.5_{0.1}}$	$86.4_{0.1}$
MultiNLI	$79.1_{0.3}$	$76.9_{0.3}$	$77.1_{0.3}$	$77.1_{0.3}$	$79.2_{0.4}$	$\mathbf{79.7_{0.5}}$	$79.3_{0.3}$
Average	81.1	80.3	80.4	80.3	81.3	81.4	**81.5**

Table 2: Comparison of a) our main multidomain model (**Multi**), b) models pretrained solely on *Pubmed* (**Pub-10/100/500**), and c) models after continued adaptation to *Pubmed* (**+Pub-10/100/500**).

5.3 Adaptation to Irrelevant Domain

To establish whether the multidomain model is indeed benefiting from exposure to multiple domains, or whether this is a case where more data means better modeling, we train a model on solely *Amazon* using as much data as the total amount of data in the main multidomain model (roughly 8GBs). We show that this model performs worse than the multidomain model (Appendix D). Thus, it is the use of multiple domains that is beneficial in this setting and not strictly the amount of data.

5.4 Domain Adaptation Order

Experiments were conducted to determine how much domain adaptation order affects performance. In our main comparisons, the domain order was *Amazon* → *Reddit Comments* → *Realnews* → *Arxiv* (*Am-RC-R-Ar*). Here we examine the accuracy of the rest of the possible adaptation orders.

The average performance of all orders across the given tasks is 81.4, with a minimum of 81.1 and a maximum of 81.7, whereas base DistilBERT scores an average of 79.3. In the worst case, there is still an improvement of 1.8, while on average we see an improvement of 2.1 over the base model.

In general, performance didn't fluctuate substantially between different orders. A plausible assumption is that the last adapted domain would have a large effect on performance, especially on tasks in that domain. This is not the case though; there seems to be no correlation between last adapted domain and task accuracy. Extensive results are presented in Appendix F.

5.5 Joint Domain Adaptation

So far we have only the case where we continuously adapt to domains in succession. After Domain A, we adapt to Domain B, then C, etc. What happens when we adapt on all domains at the same time? When training sequentially, it is plausible that a later domain will overpower an earlier one. Maybe this will be mitigated by training on all domains jointly. For this experiment, domain datasets are merged into the same training set via the following scheme: the first batch is comprised of Domain A samples, the second batch of Domain B and so on. We observe that performance remained unchanged. We showcase this experiment in Appendix E.

6 Conclusion

In this work we show that domain adaptation can be extended to multiple domains. These multidomain models are able to tackle tasks across various domains with minimal performance drop compared to single-domain models, while using fewer resources and reducing our carbon footprint.

In addition, based on Zhao et al. (2020), we can use several finetuned instances of a multidomain model for a number of tasks, with negligible increase in memory usage. So our multidomain models are also beneficial on low-resource devices.

Finally, we show that adapting to multiple domains always provides a performance increase and that tasks in low-resource domains receive a boost from multidomain models.

Acknowledgments. This work was supported by ERCAdG #740516.

References

Nick Brooks. 2018. Women's e-commerce clothing reviews. Accessed: 23-06-2020.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Arman Cohan, Waleed Ammar, Madeleine Van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. In *NAACL*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

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, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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

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.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. 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 4238–4248, Hong Kong, China. Association for Computational Linguistics.

Ruining He and Julian McAuley. 2016. Ups and downs. *Proceedings of the 25th International Conference on World Wide Web - WWW '16*.

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.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Brunak S Lund O Oprea TI Taboureau O Kringelum J, Kjaerulff SK. 2016. Chemprot-3.0: a global chemical biology diseases mapping.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning.

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.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

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.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. Inexpensive domain adaptation of pretrained language models: Case studies on biomedical ner and covid-19 qa.

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2020. Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. In *Proceedings of The 12th Language Resources* *and Evaluation Conference*, pages 4933–4941, Marseille, France. European Language Resources Association.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning.

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

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl;dr: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.

Zijian Wang and Christopher Potts. 2019. TalkDown: A corpus for condescension detection in context. 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 3711–3719, Hong Kong, China. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9054–9065. Curran Associates, Inc.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification.

Mengjie Zhao, Tao Lin, Martin Jaggi, and Hinrich Schütze. 2020. Masking as an efficient alternative to finetuning for pretrained language models.

A Task Data Overview

In *ACL-ARC* (Jurgens et al., 2018), the task is to classify citation intent in excerpts from ACL papers. *SciCite* (Cohan et al., 2019) is also a citation intent classification task, covering multiple scientific domains. *HyperPartisan* (Kiesel et al., 2019) is a news dataset, where given an article the task is to predict whether it is hyperpartisan (ie. one-sided) or not. In *AG-News* (Zhang et al., 2015), we need to predict one of four possible news topics given the article text. In *IMDB* (Maas et al., 2011) and *Clothing Reviews* (Brooks, 2018), given a review text the corresponding rating must be inferred. In *SARC* (Khodak et al., 2018), we are tasked with identifying whether a Reddit comment contains sarcasm or not. *TalkDown* (Wang and Potts, 2019) presents Reddit comment pairs and we are tasked with identifying if the reply is condescending to the original comment. *PubMed-RCT* (Dernoncourt and Lee, 2017) is a dataset containing sentences from biomedical paper abstracts alongside their role in the abstract (for example, 'background', 'result'), while for *ChemProt* (Kringelum J, 2016) we are tasked with identifying relations between proteins and chemicals.

B Domain Adaptation Hyperparameters

For domain adaptation, hyperparameters are the same for both setups (single-domain and multidomain), across all domains. Minimal hyperparameter tuning was performed, with the main goal being to keep training as computationally efficient as possible. Batch size and sequence length were set to 32 and the learning rate to $1e$-5. Models were trained for a single epoch. The multidomain model was therefore trained for 4 epochs in total, one for each domain.

C Supervised Task Learning Hyperparameters

Apart from the difference in epochs and the number of classifier neurons, all other hyperparameters are the same for all models during supervised task learning. Learning rate was kept at $4e$-5, as suggested in Devlin et al. (2019). Maximum sequence length was set to 128 while we used 32 batches for training and testing. All hyperparameters were selected upon evaluation on the development sets.

D Irrelevant Domain Adaptation Results

Results for adaptation to an irrelevant domain. In this case, DistilBert is pretrained entirely on *Amazon* (8GBs). As in Gururangan et al. (2020), we find that when adapting to an irrelevant domain, performance does not increase.

	Multi	**Amazon**
ACL-ARC	$\mathbf{70.5_{2.1}}$	$68.2_{1.3}$
SciCite	$\mathbf{86.1_{0.2}}$	$85.1_{0.3}$
SARC	$\mathbf{76.1_{0.3}}$	$75.3_{0.2}$
TalkDown	$\mathbf{86.1_{0.7}}$	$\mathbf{86.1_{0.5}}$
HyperPartisan	$\mathbf{80.5_{1.4}}$	$79.3_{1.1}$
AG-News	$\mathbf{94.3_{0.1}}$	$94.1_{0.1}$
IMDB	$\mathbf{86.7_{0.1}}$	$\mathbf{86.7_{0.1}}$
Clothing	$69.3_{0.3}$	$\mathbf{69.5_{0.1}}$
ChemProt	$77.0_{0.6}$	$\mathbf{77.4_{0.8}}$
Pubmed-RCT	$\mathbf{86.4_{0.1}}$	$86.1_{0.1}$
MultiNLI	$\mathbf{79.1_{0.3}}$	$77.2_{0.3}$
Average	**81.1**	80.4

Table 3: Comparison between our main model (**Multi**) and a model pretrained entirely on **Amazon** data.

E Joint Domain Adaptation Results

Here we compare results between sequential (*Am-RC-R-Ar*) and joint domain adaptation. Results remain similar, showing that there is no substantial difference between the two adaptation methods.

	Sequential	**Joint**
ACL-ARC	$70.5_{2.1}$	$\mathbf{72.6_{1.9}}$
SciCite	$86.1_{0.2}$	$\mathbf{86.5_{0.2}}$
SARC	$\mathbf{76.1_{0.3}}$	$76.0_{0.2}$
TalkDown	$\mathbf{86.1_{0.7}}$	$86.0_{0.6}$
HyperPartisan	$80.5_{1.4}$	$\mathbf{80.9_{1.2}}$
AG-News	$\mathbf{94.3_{0.1}}$	$94.2_{0.1}$
IMDB	$\mathbf{86.7_{0.1}}$	$\mathbf{86.7_{0.1}}$
Clothing	$69.3_{0.3}$	$\mathbf{69.4_{0.2}}$
ChemProt	$77.0_{0.6}$	$\mathbf{77.9_{0.7}}$
Pubmed-RCT	$\mathbf{86.4_{0.1}}$	$\mathbf{86.4_{0.1}}$
MultiNLI	$79.1_{0.3}$	$\mathbf{79.2_{0.3}}$
Average	81.1	**81.4**

Table 4: Comparison of sequential and joint models.

F Domain Adaptation Order Results

Results for adaptation order experiments are given in Tables 5, 6, 7 and 8 (alphabetical order).

	Am-Ar-R-RC	**Am-Ar-RC-R**	**Am-R-Ar-RC**	**Am-R-RC-Ar**	**Am-RC-Ar-R**	**Am-RC-R-Ar**
ACL-ARC	$74.1_{1.2}$	$71.0_{2.4}$	$72.7_{3.0}$	$75.1_{0.3}$	$72.7_{2.1}$	$70.5_{1.0}$
SciCite	$86.0_{0.3}$	$86.3_{0.1}$	$85.7_{0.5}$	$85.9_{0.3}$	$85.5_{0.2}$	$86.1_{0.2}$
SARC	$75.9_{0.1}$	$76.2_{0.1}$	$76.0_{0.3}$	$75.9_{0.2}$	$75.7_{0.3}$	$76.1_{0.1}$
TalkDown	$86.4_{0.5}$	$86.1_{1.2}$	$85.9_{0.6}$	$86.1_{1.0}$	$85.9_{0.7}$	$86.1_{0.6}$
HyperPartisan	$80.1_{2.5}$	$78.4_{0.7}$	$80.5_{3.8}$	$81.6_{1.3}$	$81.3_{1.4}$	$80.5_{1.7}$
AG-News	$94.2_{0.2}$	$94.4_{0.1}$	$94.3_{0.1}$	$94.2_{0.0}$	$94.5_{0.1}$	$94.3_{0.1}$
IMDB	$86.5_{0.1}$	$86.6_{0.1}$	$86.7_{0.1}$	$86.6_{0.1}$	$86.6_{0.2}$	$86.7_{0.1}$
Clothing	$69.2_{0.2}$	$69.4_{0.4}$	$69.4_{0.4}$	$69.2_{0.1}$	$69.5_{0.3}$	$69.3_{0.4}$
ChemProt	$78.2_{0.3}$	$77.7_{0.3}$	$77.5_{0.5}$	$77.7_{0.8}$	$77.7_{0.6}$	$77.0_{0.3}$
Pubmed-RCT	$86.4_{0.1}$	$86.4_{0.0}$	$86.5_{0.1}$	$86.5_{0.1}$	$86.4_{0.1}$	$86.4_{0.1}$
MultiNLI	$79.3_{0.2}$	$79.1_{0.3}$	$79.3_{0.3}$	$79.2_{0.1}$	$79.3_{0.3}$	$79.1_{0.2}$
Average	81.5	81.1	81.3	81.6	81.4	81.1

Table 5: Domain adaptation order comparison - Orders commencing with *Amazon*.

	Ar-Am-R-RC	**Ar-Am-RC-R**	**Ar-R-Am-RC**	**Ar-R-RC-Am**	**Ar-RC-Am-R**	**Ar-RC-R-Am**
ACL-ARC	$72.9_{0.7}$	$70.5_{1.2}$	$72.9_{1.5}$	$71.2_{1.6}$	$74.8_{1.8}$	$72.4_{1.4}$
SciCite	$86.1_{0.1}$	$86.5_{0.5}$	$85.8_{0.2}$	$85.9_{0.4}$	$86.1_{0.2}$	$86.2_{0.3}$
SARC	$75.9_{0.3}$	$76.2_{0.2}$	$76.1_{0.1}$	$75.8_{0.4}$	$76.1_{0.2}$	$76.0_{0.3}$
TalkDown	$86.3_{0.3}$	$86.3_{0.3}$	$85.9_{0.8}$	$86.7_{0.3}$	$85.9_{0.9}$	$86.2_{0.7}$
HyperPartisan	$81.5_{2.1}$	$82.1_{1.5}$	$82.7_{1.5}$	$80.3_{1.9}$	$78.5_{1.8}$	$79.3_{2.3}$
AG-News	$94.4_{0.2}$	$94.3_{0.1}$	$94.2_{0.2}$	$94.3_{0.2}$	$94.4_{0.1}$	$94.3_{0.1}$
IMDB	$86.6_{0.2}$	$86.4_{0.0}$	$86.6_{0.1}$	$86.6_{0.2}$	$86.6_{0.2}$	$86.7_{0.2}$
Clothing	$69.6_{0.3}$	$69.3_{0.2}$	$69.5_{0.5}$	$69.7_{0.4}$	$69.3_{0.2}$	$69.5_{0.2}$
ChemProt	$77.7_{0.3}$	$78.3_{0.4}$	$77.7_{0.8}$	$78.9_{0.4}$	$78.3_{0.7}$	$77.4_{1.2}$
Pubmed-RCT	$86.5_{0.1}$	$86.4_{0.1}$	$86.5_{0.1}$	$86.5_{0.1}$	$86.5_{0.1}$	$86.5_{0.0}$
MultiNLI	$78.9_{0.3}$	$79.4_{0.2}$	$79.2_{0.2}$	$79.3_{0.3}$	$79.4_{0.3}$	$79.0_{0.1}$
Average	81.5	81.4	81.6	81.4	81.5	81.2

Table 6: Domain adaptation order comparison - Orders commencing with *Arxiv*.

	R-Am-Ar-RC	**R-Am-RC-Ar**	**R-Ar-Am-RC**	**R-Ar-RC-Am**	**R-RC-Am-Ar**	**R-RC-Ar-Am**
ACL-ARC	$74.8_{1.7}$	$72.4_{1.3}$	$70.7_{2.4}$	$72.2_{2.9}$	$73.1_{3.2}$	$72.7_{1.2}$
SciCite	$85.8_{0.4}$	$85.6_{0.4}$	$86.2_{0.1}$	$86.0_{0.3}$	$85.9_{0.4}$	$86.3_{0.2}$
SARC	$76.4_{0.3}$	$76.3_{0.2}$	$75.9_{0.3}$	$75.9_{0.3}$	$76.1_{0.1}$	$75.8_{0.4}$
TalkDown	$86.3_{0.2}$	$86.4_{0.1}$	$86.4_{0.7}$	$85.9_{0.4}$	$86.7_{0.4}$	$85.8_{0.9}$
HyperPartisan	$81.5_{2.1}$	$83.5_{2.9}$	$80.4_{1.2}$	$80.6_{1.3}$	$81.0_{3.1}$	$81.3_{1.8}$
AG-News	$94.2_{0.1}$	$94.2_{0.1}$	$94.4_{0.1}$	$94.2_{0.1}$	$94.3_{0.1}$	$94.4_{0.1}$
IMDB	$86.7_{0.0}$	$86.7_{0.1}$	$86.4_{0.2}$	$86.6_{0.0}$	$86.6_{0.1}$	$86.7_{1.5}$
Clothing	$69.7_{0.2}$	$69.2_{0.2}$	$69.6_{0.1}$	$68.8_{0.6}$	$69.2_{0.3}$	$68.8_{0.3}$
ChemProt	$77.8_{0.8}$	$77.1_{1.0}$	$77.9_{0.9}$	$77.9_{0.8}$	$79.0_{1.0}$	$79.0_{0.3}$
Pubmed-RCT	$86.5_{0.1}$	$86.4_{0.1}$	$86.5_{0.1}$	$86.5_{0.1}$	$86.5_{0.1}$	$86.4_{0.0}$
MultiNLI	$79.1_{0.1}$	$79.2_{0.2}$	$79.1_{0.2}$	$79.4_{0.3}$	$79.0_{0.4}$	$78.8_{0.2}$
Average	81.7	81.6	81.2	81.3	81.6	81.5

Table 7: Domain adaptation order comparison - Orders commencing with *Realnews*.

	RC-Am-Ar-R	**RC-Am-R-Ar**	**RC-Ar-Am-R**	**RC-Ar-R-Am**	**RC-R-Am-Ar**	**RC-R-Ar-Am**
ACL-ARC	$70.7_{1.2}$	$72.9_{2.5}$	$72.0_{1.0}$	$71.0_{3.4}$	$76.0_{2.1}$	$74.3_{1.2}$
SciCite	$86.5_{0.1}$	$86.4_{0.2}$	$85.9_{0.3}$	$86.2_{0.2}$	$86.0_{0.2}$	$86.0_{0.3}$
SARC	$76.2_{0.2}$	$75.9_{0.3}$	$76.3_{0.1}$	$75.9_{0.3}$	$76.1_{0.3}$	$76.3_{0.3}$
TalkDown	$86.1_{0.5}$	$86.6_{0.3}$	$85.9_{1.0}$	$86.7_{0.8}$	$86.0_{0.5}$	$86.1_{0.4}$
HyperPartisan	$82.3_{2.0}$	$82.1_{1.6}$	$80.9_{2.3}$	$80.6_{0.9}$	$81.2_{1.3}$	$78.8_{1.4}$
AG-News	$94.1_{0.1}$	$94.2_{0.1}$	$94.1_{0.1}$	$94.4_{0.1}$	$94.4_{0.1}$	$94.3_{0.2}$
IMDB	$86.6_{0.2}$	$86.7_{0.2}$	$86.7_{1.5}$	$86.7_{1.5}$	$86.7_{1.5}$	$86.4_{0.2}$
Clothing	$69.5_{0.4}$	$69.9_{0.5}$	$69.3_{0.7}$	$69.5_{0.4}$	$69.1_{0.1}$	$69.1_{0.4}$
ChemProt	$79.2_{0.4}$	$78.3_{0.6}$	$78.6_{0.3}$	$78.7_{0.9}$	$77.9_{0.5}$	$78.0_{0.9}$
Pubmed-RCT	$86.5_{0.1}$	$86.4_{0.1}$	$86.5_{0.1}$	$86.5_{0.1}$	$86.4_{0.1}$	$86.4_{0.1}$
MultiNLI	$79.2_{0.4}$	$79.3_{0.1}$	$79.3_{0.2}$	$79.1_{0.2}$	$79.0_{0.4}$	$78.9_{0.3}$
Average	81.5	81.7	81.4	81.4	81.7	81.3

Table 8: Domain adaptation order comparison - Orders commencing with *Reddit Comments*.