Modular Domain Adaptation

Anonymous ACL submission

Abstract

Off-the-shelf models are widely used by computational social science researchers to measure properties of text, such as sentiment. However, without access to source data it is difficult to account for domain shift, which presents a threat to validity. Here, we treat domain adaptation as a modular process that involves separate model producers and model consumers, and show how they can independently cooperate to facilitate more accurate measurements of text. We introduce two lightweight techniques for this scenario, and demonstrate that they reliably increase out-of-domain accuracy on four multi-domain text classification datasets when used with linear and contextual embedding models. We conclude with recommendations for model producers and consumers, and release models and replication code to accompany this paper.

1 Introduction

Machine learning models for tasks like sentiment analysis and hate speech detection are becoming increasingly ubiquitous as off-the-shelf tools, including as commercial packages or cloud-based APIs. Among other applications, these models are widely used by computational social scientists to obtain standardized measurements of various document properties at scale. However, the problem of domain shift represents a threat to validity, one which is difficult for practitioners to overcome, especially without access to source data—which may be unavailable for reasons of privacy, copyright, or commercial interests. In this paper, we propose to treat domain adaptation as a *modular* process involving both *model producers* and *model consumers*, and show how both parties can independently cooperate to produce more reliable measurements.

Although this framework applies to any application involving independent model producers and consumers, we focus here on text-based instruments, including both lexicons and supervised text classification models. Using multiple datasets and baselines, we show that model consumers can obtain more accurate results by using models designed to be lightly adapted, and that model producers can facilitate such adaptation, even without providing access to source data, using what we call *anticipatory domain adaptation* (see Figure 1).

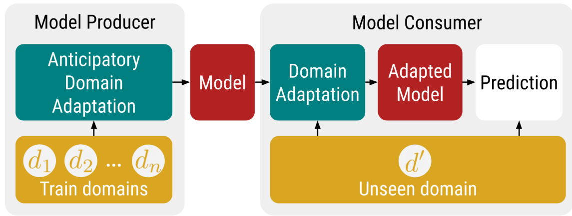

Figure 1: Modular domain adaptation involves both model producers and model consumers, cooperating via a standardized model.

We introduce two techniques under this new paradigm: domain-specific bias (DsBias) and domain-specific normalization (DsNorm). These methods enable model consumers to incorporate information from their domain of interest—without additional training or hyperparameter tuning—and provide reliably better out-of-domain accuracy for both linear and contextual embedding classifiers.

In summary, this paper makes the following contributions:

- We present *modular domain adaptation* as a process that involves both model producers and model consumers (§3.1).
- We introduce two simple techniques for *anticipatory domain adaptation* – that is, ways in which model producers can facilitate adaptation by model consumers (§3.4).
- We quantify the relative out-of-domain performance of linear and contextual embedding models in combination with various adaptation techniques on multiple datasets (§4).

- We release linear and contextual models for measuring *framing* in text based on the Media Frames Corpus (Card et al., 2015).[1]

2 Background and Related Work

There is an extensive literature on using text as data in computational social science (CSS) to study political communication, mental health, and many other social phenomena (Grimmer and Stewart, 2013; Fulgoni et al., 2016; Eichstaedt et al., 2018; Saha et al., 2019; Li et al., 2020b; Jaidka et al., 2020; Nguyen et al., 2020). The overarching requirement in much of this work is to convert raw text (from speeches, articles, tweets, etc.) into a quantitative representation capturing some property of interest, such as sentiment or affect (Hatzivassiloglou and McKeown, 1997; Huettner and Subasic, 2000; Hutto and Gilbert, 2014). Although some researchers develop bespoke models for specialized applications, those studying similar phenomena often make use of a shared set of tools, in principle allowing for comparison across studies.

Among the most commonly used instruments are lexicons such as LIWC (Tausczik and Pennebaker, 2010), EmoLex (Mohammad and Turney, 2013), and the moral foundations dictionary (Frimer et al., 2019), which offer simple, reproducible, and interpretable measurements, despite being insensitive to context.[2] Although lexicons are often developed without the use of machine learning, we can treat them interchangeably with linear models, as they are typically utilized by summing the presence of the listed features (i.e., words). The output of such models is thus a score for each document, allowing for comparisons between groups of documents, such as across time, sources, or treatment groups. Importantly, these scores should be thought of as proxies for theoretical constructs of interest, such as sentiment or ideology, to which they provide a noisy approximation (Jacobs and Wallach, 2021; Pryzant et al., 2021).[3]

Although open source models have numerous advantages for research, model creators may be unable or unwilling to share the data that their models are based on, especially for commercial lexicons, like LIWC, and cloud-based products like Perspective API. Despite their limitations, these systems provide convenient, comparable, and easy-to-use tools for CSS researchers in various fields. However, those who use such models face the dual problems of adapting them to a new domain and assessing validity in that domain, and will often want to do so with relatively constrained resources.

Domain adaptation is an important area of research within machine learning, but most work tends to assume either access to source data (e.g., for re-weighting; Huang et al., 2007; Jiang and Zhai, 2007; Azizzadenesheli et al., 2019), or extensive labeled data in the new domain. For contextual embedding models in NLP, continued training on a small amount of labeled data offers benefits (Radford et al., 2017; Howard and Ruder, 2018), though this requires sufficient data for fine-tuning, validation, and evaluation (to assess performance in the target domain), as well access to sufficient computational resources (typically GPUs).

Self-training (augmenting source data using predicted labels in the new domain) provides an alternative strategy, and has shown to work both theoretically and practically (Kumar et al., 2020), but typically assumes access to the original source data, and requires making choices about multiple hyperparameters, which is difficult in the absence of extensive validation data. A few papers have considered the problem of domain adaptation without source data (Chidlovskii et al., 2016; Liang et al., 2020), but most tend to emphasize resource-intensive solutions (e.g., using GANs; Li et al., 2020a).

A different but related paradigm is "deconfounded lexicon induction" (Pryzant et al., 2018a,b), where the goal is to learn a model that accounts for the influence of non-textual attributes (such as domain). Because this approach tries to eliminate the influence of confounders, we might expect it to produce a more domain-agnostic model, and we therefore include experiments with the proposed techniques for the purpose of comparison.

3 Methods

3.1 Problem Formulation

In this work, we make the distinction between *model producers* and *model consumers*. Model producers wish to train a model on a labeled dataset of documents coming from one or more domains (e.g., political issues, or paper categories), where

[1]To be released after review period for anonymity.

[2]In this paper, we use "lexicon" to refer to weighted or unweighted list of words corresponding to categories of interest.

[3]Although lexicons are often used to obtain real-valued scores, rather than as classifiers, we assume for the sake of simplicity that any available in-domain annotations are collected as categorical labels, and evaluate all models as classifiers, using an appropriate threshold where necessary.

each document, $\mathbf{x}_i$, has an associated categorical class label, $y_i \in \mathcal{Y}$, as well as a domain, $d_i \in \mathcal{D}$. Model consumers, by contrast, will apply the trained model to a new domain, $d' \notin \mathcal{D}$, without access to either the source data or extensive labeled data from their domain of interest.[4]

Note that in our setup, the producer and consumer have different goals and face different constraints. The model producer's goal is to create a self-contained model, without sharing any source data associated with training, due to reasons such as privacy, copyright, or commercial interests. The model consumer's goal, by contrast, is to achieve high accuracy in a new domain, d', without needing extensive resources for either labeling data or training a new model. Especially for applications in CSS, we also assume that model consumers will need to estimate accuracy in their domain, as part of demonstrating validity (Jacobs and Wallach, 2021).

In this paper, we compare the performance under these constraints of two especially common approaches to creating text classification models (logistic regression with bag-of-words features and contextual embedding models), and propose two methods (DsBiasand DsNorm; §3.4) by which model producers can facilitate domain adaptation by model consumers.

3.2 Underlying Models

As foundations from which to experiment with techniques for modular domain adaptation, we make use of two standard baseline approaches in text classification: regularized logistic regression and fine-tuned contextual embedding models. In both cases, the model is trained using an appropriate loss function (e.g., logistic or cross entropy), computed with respect to predicted probabilities:

$$\hat{\mathbf{p}}_i = \mathrm{softmax}(\mathbf{b} + f(\mathbf{x}_i)^\top \mathbf{W}) \qquad (1)$$

where $\mathbf{b} \in \mathbb{R}^k$ is a bias vector, $\mathbf{W}$ is an $h \times k$ weight matrix, $f(\cdot)$ encodes a document as an h-dimensional vector, and $\hat{\mathbf{p}}_i \in \Delta^k$ is the predicted distribution over k classes.[5]

For logistic regression, $f(\cdot)$ encodes $\mathbf{x}_i$ as a sparse bag-of-words vector, with h equal to the size of the vocabulary. For contextual embedding

[4] We assume that typical model consumers in CSS are capable of generating some labeled data in their domain (e.g., by manually annotating data), but have insufficient resources available to create a large labeled dataset.

[5] Or equivalently for binary labels: a logistic function instead of a softmax, $p_i \in [0, 1]$, $b \in \mathbb{R}$, and $\mathbf{w} \in \mathbb{R}^h$.

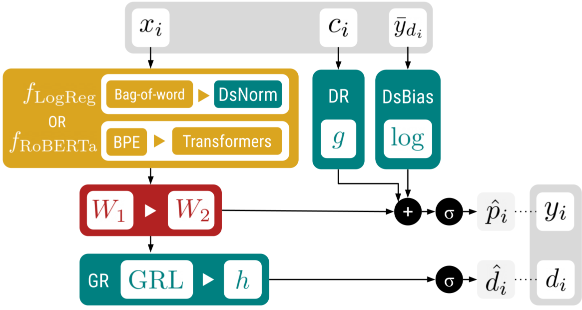

Figure 2: Model diagrams of base predictors in conjunction with proposed techniques, showing how pieces fit together. All deconfounding and adaptation techniques are marked in green and are optional. Base predictor is marked in yellow.

models, $f(\mathbf{x}_i) \in \mathbb{R}^h$ is the penultimate dense representation produced by feeding document i into a contextual embedding model, plus additional layers in the case of a multi-layer decoder.

3.3 Deconfounding Techniques

To augment the underlying models, we begin with previously proposed techniques for removing the influence of domain. Although mainly designed to account for explicitly modeled features of the data, and not specifically focused on domain adaptation, Pryzant et al. (2018b) proposed two methods for *deconfounded lexicon induction*—that is, attenuating the influence of non-textual document properties, including domain, when learning an interpretable model. Since these are carried out solely by model producers, we use utilize them as baselines.

Deep Residualization (DR): As one way of deconfounding labels from potential confounds, Pryzant et al. (2018b) proposed learning a mapping from observable confounds to labels, and integrating that into the prediction. Specifically, we replace the bias term $\mathbf{b}$ in Eq. (1) with an instance specific vector, i.e.,

$$\hat{\mathbf{p}}_i = \mathrm{softmax}(g(\mathbf{c}_i) + f(\mathbf{x}_i)^\top \mathbf{W}), \qquad (2)$$

where $\mathbf{c}_i$ is a vector of confounds for document i, and $g(\cdot)$ is a feed-forward network mapping from confounds to a dense vector representation $\in \mathbb{R}^k$.

In our case, $\mathbf{c}_i$ is a one-hot vector representing domain (i.e., d_i). Since the ultimate application domain is not available at training time, the model consumer would use the domain agnostic predictor, setting $g(\mathbf{c}_i) = \mathbf{0}$ for the unseen domain.

Gradient Reversal (GR): Pryzant et al. (2018b) also proposed using gradient reversal for deconfounding. That is, we train the model to successfully predict an instance's label, while being *unable* to predict the domain. To implement this, we factorize the weight matrix $\mathbf{W}$ into two matrices, $\mathbf{W}_1$ and $\mathbf{W}_2$, and apply gradient reversal to the intermediate representation, i.e.

$$\hat{\mathbf{p}}_i = \text{softmax}(\mathbf{b} + (f(\mathbf{x}_i)^\top \mathbf{W}_1)^\top \mathbf{W}_2) \quad (3)$$

$$\hat{\mathbf{d}}_i = \text{softmax}(h(\text{GRL}(f(\mathbf{x}_i)^\top \mathbf{W}_1))), \quad (4)$$

where $\hat{\mathbf{d}}_i \in \Delta^{|\mathcal{D}|}$ is the predicted distribution over domains, $h(\cdot)$ is a feed-forward network, and GRL reverses the gradients with respect to $\mathbf{W}_1$ during training (Ganin et al., 2016).

3.4 Anticipatory Adaptation Techniques

As mentioned, the above techniques were designed for deconfounding by the model producer, and not for domain adaptation by the model consumer. Here we introduce two new methods by which a model producer might facilitate adaptation, without having to share training data or requiring knowledge of the model consumer's domain.

Domain-Specific Bias (DSBIAS): A key limitation of deep residualization (DR) is that it has no way to incorporate information about a previously unseen domain. As an alternative, we modify the idea of DR by expressing the instance-specific bias in terms of the distribution of labels in the corresponding domain. This allows model consumers to inject information about a new domain into the model at prediction time, given knowledge about the relevant label distribution. Specifically, for each domain d we set the bias term in Eq. (1) to be the element-wise log of a vector of label frequencies in that domain, i.e.,

$$\hat{\mathbf{p}}_i = \text{softmax}(\log(\bar{\mathbf{y}}_{d_i}) + f(\mathbf{x}_i)^\top \mathbf{W}) \quad (5)$$

where $\bar{\mathbf{y}}_{d_i} \in \Delta^k$ is a vector of estimated label frequencies in the domain of instance i. Using the log of the estimated label frequencies means that the learned weights ($\mathbf{W}$) represent additive deviations (in log space) from baseline frequencies, much like in SAGE (Eisenstein et al., 2011).

At training time, $\bar{\mathbf{y}}_{d_i}$ can be estimated by the model producer from labeled data in each domain. At prediction time, model consumers can provide an approximate label distribution for a new domain by either estimating it from a small amount of labeled data, or by leveraging prior knowledge of the domain itself. Thus, DSBIAS benefits from having some labeled data in the new domain, but does not require additional training by model consumers.

Domain-Specific Normalization (DSNORM): As an additional option for linear models, and inspired by normalization techniques used in deep learning, we also consider normalizing each element in the bag-of-words feature vector according to its expected frequency of the individual domain:

$$f'(\mathbf{x}_i) = f(\mathbf{x}_i) - \Sigma_{j=1}^{N_{d_i}} f(\mathbf{x}_j)/N_{d_i}, \quad (6)$$

where $f(\mathbf{x}_i)$ is a vector of feature values, and N_{d_i} is the number of instances in the domain of instance i. This allows for a commonly occurring word (e.g., the word "climate" in climate change news) to become less important if it occurs in the current domain, and relatively more important in others.[6] Because this does not require labeled data, it can be applied directly to a new domain by model consumers.

3.5 Domain Fine-Tuning (DFT)

Past work on pretrained contextual embedding models has demonstrated that continued training on labeled samples from a new domain can effectively adapt the model to that domain, improving performance (Radford et al., 2017; Howard and Ruder, 2018; Gururangan et al., 2020). Although powerful, there are several reasons why this may not be an option for model consumers. First, many APIs and commercial systems will not provide this functionality or expose the necessary parts of the model. Second, the computational resources required for fine-tuning (i.e., GPUs) may be prohibitive for some users. Third, fine tuning means that individual model consumers will no longer be applying the same standardized model, thus reducing the comparability of results. Nevertheless, we include experiments with DFT in order to quantify how much better a model consumer could do with sufficient labeled data for training and evaluation in their domain (§4), and compare fine tuning an off-the-shelf model to one that has been fine-tuned for the same task on out-of-domain data (§4.5).

4 Experiments

In this section we systematically evaluate the performance of both underlying models in conjunction

[6]Like TF-IDF, DSN scales feature values based on frequency, but keeps all (binarized) feature values between 0 and 1, even for rare words.

Dataset	$\|\mathcal{Y}\|$	Domains	Min N_d	Max N_d
MFC	15	6	4220	8898
ARXIV	4	6	5338	59612
AMAZON	3	5	4199	22573
SENTI	2	5	3088	10003

Table 1: Dataset statistics, showing the number of categories (labels), domains, and minimum and maximum number of labeled instances per domain. For details of data splits, see appendix G.

with all available techniques in section §3, to quantitatively evaluate their performance, and to derive best practices as advice to practitioners when applying them to real data under various settings. For simplicity, we use accuracy as the primary metric of evaluation in all our experiments.

4.1 Data

Because our primary interest is to evaluate modular domain adaptation techniques, we choose datasets with instances from multiple known domains, so that we can hold out each domain in turn to estimate performance when adapting to a previously unseen domain. In particular, we make use of four datasets in our experiments (see Table 1): the Media Frame Corpus (MFC; Card et al., 2015) and the arXiv Dataset (ARXIV; Clement et al., 2019), the Amazon Reviews Dataset (AMAZON; Ni et al., 2019), and a collection of sentiment classification datasets (SENTI; see below).

MFC is a dataset of news articles on 6 different issues (e.g., "climate change"), and each article is labeled to have 1 of 15 possible primary "frames", which are assumed to generalize across issues. As intuition would suggest, different frames are emphasized in coverage of different issues (e.g., climate change is discussed more in terms of "capacity and resources" than "crime and punishment").

ARXIV is the dataset of all scholarly articles published on `arXiv.org`. We consider articles in 6 categories in the taxonomy relevant to machine learning (e.g., cs.CL, "Computation and Language"). For each article, we consider the year in which it was published, discretised into 4 time periods, and try to predict the time period from the abstract, using taxonomic categories as domains.[7]

AMAZON is a subsampled dataset of product reviews from Amazon from the most popular 7 categories. Each review is associated with a review score (negative: 1; neutral: 2-4; positive: 5) which we try to predict from the review text.

[7]Divided by the years 2008, 2014, and 2019, which are rough markers of major machine learning milestones.

SENTI is a collection of diverse, subsampled sentiment classification datasets: Twitter US Airline Sentiment (Eight, 2015), Amazon Books Reviews (Ni et al., 2019), IMDb Movie Reviews (Maas et al., 2011), Sentiment 140 Tweets (Go et al., 2009), and the Stanford Sentiment Treebank (SST; Socher et al., 2013). The domains included in this dataset differ from each other in various ways (e.g., IMDb reviews are often a few paragraphs long, whereas SST utterances are much shorter), which is intended to mimic scenarios in which users might apply off-the-shelf sentiment analysis tools. From each sample we classify instances as positive or negative.

4.2 Implementation Details

As a linear baseline, we use L1-regularized logistic regression (LogReg) operating on binarized bag of word features, which has been shown to be a competitive choice among similar models (Wang and Manning, 2012). We limit ourselves to a vocabulary of the 5000 most frequent lowercased words in the training set. We use full-batch gradient descent to optimize the models, with L1 regularization on the weight matrices only. Regularization strength is determined for each configuration using grid search on in-domain cross validation splits, then applied to the full in-domain training set.

For contextual embedding classifiers, we use RoBERTa, fine-tuning the publicly available `roberta-base` from Hugging Face (Wolf et al., 2020), using AdamW (Loshchilov and Hutter, 2019) with a fixed dropout rate of 0.2. We use early stopping with number of epochs determined for each configuration using in-domain cross validation splits, then applied to the full in-domain training set. For additional details, please refer to Appendix I.

4.3 Out-of-domain Performance

As our primary evaluation, we assess each technique in combination with each of our base models (LogReg vs. RoBERTa). For each domain of each dataset, we create a dedicated held-out test set. During training, for each dataset, we hold out each domain in turn, and use the remaining domains as in-domain training data. We report average performance on out-of-domain test sets, along with variance (across domains) in improvement over the baseline model in Table 2. For DsBias, we

		MFC		ARXIV		AMAZON		SENTI	
		acc	σ_Δ	acc	σ_Δ	acc	σ_Δ	acc	σ_Δ
	Most common	0.276	-	0.526	-	0.631	-	0.495	-
LogReg	Base	0.508	-	0.543	-	0.672	-	0.647	-
LogReg	DR	0.503	0.009	0.551	0.005	0.674	0.004	0.648	0.003
LogReg	GR	0.500	0.004	0.541	0.005	0.709	0.001	0.638	0.003
LogReg	DsBias (250)	0.515	0.020	0.564	0.024	0.714	0.004	0.690	0.052
LogReg	DsNorm+DsBias (250)	0.532	0.018	0.568	0.013	0.716	0.006	0.700	0.041
LogReg	DsBias (oracle)	0.524	0.022	0.563	0.013	0.715	0.003	0.695	0.041
LogReg	DsNorm+DsBias (oracle)	0.541	0.015	0.568	0.012	0.717	0.002	0.709	0.039
RoBERTa	Base	0.599	-	0.584	-	0.772	-	0.789	-
RoBERTa	DR	0.594	0.014	0.593	0.007	0.782	0.017	0.817	0.012
RoBERTa	GR	0.202	0.039	0.512	0.003	0.777	0.012	0.684	0.068
RoBERTa	DsBias (250)	0.613	0.030	0.599	0.010	0.772	0.036	0.819	0.016
RoBERTa	DFT (250)	0.683	0.032	0.615	0.012	0.785	0.025	0.831	0.018
RoBERTa	DsBias (oracle)	0.622	0.026	0.600	0.013	0.779	0.012	0.819	0.014

Table 2: Average out-of-domain accuracy on four datasets show consistent findings for both LogReg and RoBERTa: (1) DsBias with the oracle label distribution offers a small but reliable gain in accuracy over the Base models; (2) gains are almost as large when approximating the oracle distribution with 250 labeled examples; (3) DsNorm also offers a small but reliable benefit for linear models when used in combination with DsBias; (4) Deconfounding techniques (DR and GR) do not improve out-of-domain accuracy over Base; (5) RoBERTa achieves much better out-of-domain accuracy than LogReg, even without fine tuning to the target domain; (6) Additional fine tuning to 250 labeled example (DFT) offers additional gains, though this may not be an option for some model consumers. σ_Δ is the standard deviation (across held-out domains) of the improvement over the baseline (Base).

evaluate performance both when assuming oracle knowledge of the label distribution in the held-out domain, and when we estimate it from a random sample of 250 instances, which we also use for DFT.

There are four important takeaways from these results. First, RoBERTa offers a dramatic improvement over base logistic regression in out-of-domain performance (8-18% improvement), even without additional fine-tuning by the model consumer.[8] Thus, although some model consumers may still prefer linear models or lexicons for greater interpretability (see Appendix F), the CSS community would greatly benefit from having model producers release both linear *and* contextual embedding models. Moreover, fine-tuning RoBERTa to even a small amount of in-domain labeled data produces another additional improvements (though with caveats, which we discuss in §4.5).

Second, the deconfounding techniques (DR and GR) offer little or no benefit over the baseline in terms of out-of-domain performance. Thus, while they may work for removing the influence of domain in constructing a lexicon, they do not appear to produce a domain agnostic lexicon in a way that is beneficial for model consumers.

Third, DsBias (using the log label distribution for each domain) offers a small but reliable benefit (2-4%) to model consumers when working with a known label distribution, and this applies to both linear and contextual embedding models. Moreover, this still holds when model consumers estimate this distribution from a small amount of labeled data (here 250 instances). A key advantage to DsBias is that it requires no additional training by model consumers, and essentially keeps the underlying model unchanged, preserving comparability across studies. Moreover, estimating a low-dimensional label distribution requires relatively few samples, with statistically bounded errors given a random sample (see §4.4 below).

Fourth, DsNorm (normalizing features by domain) offers a small additional benefit when used in combination with DsBias for linear models, and it can be applied by model consumers based purely on unlabeled data from their domain.

Based on what evaluations can be justified using a simple power analysis (Card et al., 2020), we verify that LogReg+DsBias+DsNorm is significantly better than LogReg using McNemar's test, as is RoBERTa+DsBias compared to RoBERTa (see Appendix J).

To ensure that our linear classifiers achieve rea-

[8] As expected, both LogReg and RoBERTa show large drops in performance from the domains in which they were trained (3-10% on average, depending on dataset; see Table 6 in Appendix D).

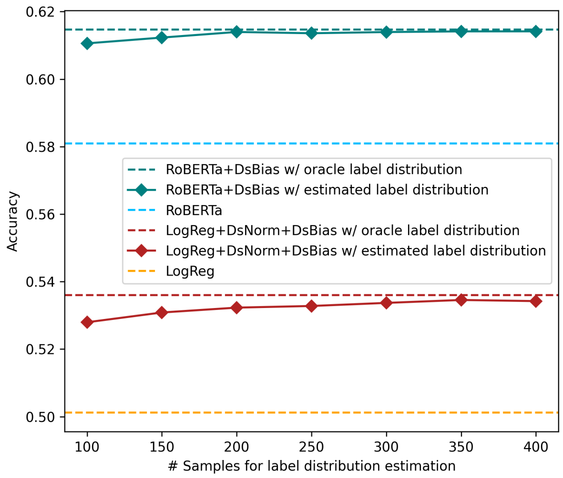

Figure 3: Average validation accuracy of in unseen domains of MFC, using a varying number of target domain samples to estimate label distribution for DsBias.

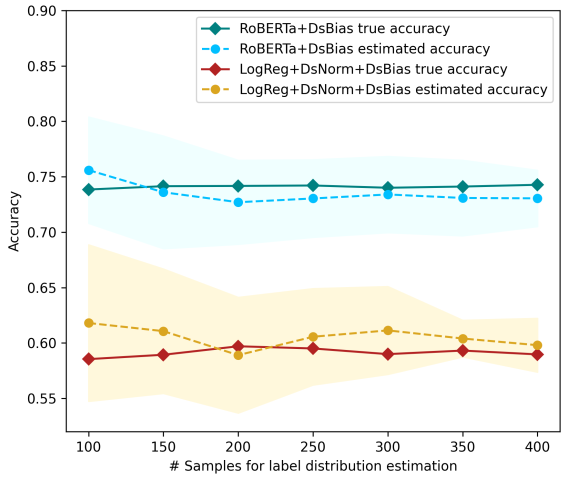

Figure 4: Validation accuracy of calculated from all holdout samples, and from limited samples, of the Sentiment 140 dataset in SENTI. Shaded area denotes 1 standard deviation from mean estimated performance. For all domains in all datasets, see appendix E.

sonable performance, we also compare our results on the SENTI dataset to several off-the-shelf sentiment lexicons, evaluating them as classifiers with fine-tuned classification thresholds, and find that none do as well as our best logistic regression classifier in terms of out-of-domain performance (see Appendix B). Finally, in Appendix C, we verify that our findings hold even if the model producer is only able to train on a single domain.

4.4 Estimating the Label Distribution

DsBias achieved the best performance when given the oracle label distribution of the target domain, but in practice this is unlikely to be known precisely. To study the effect of using an estimated label distribution with the technique, we here assume that we only have very few labeled samples from the unseen domain. Specifically, we run the same experiment in §4.3 where we vary the number of samples used to estimate the label distribution in the target domain.

Figure 3 demonstrates that with only as few as 100 labeled samples, average performance using DsBias improves from the base model, and arrives within 1 percent of accuracy from using the ground truth distribution. For each heldout domain, we run 5 trials each estimating label distribution using a fixed number of random samples, evaluate performance on the full train set of the heldout domain, then average across all trials and all heldout domains. Further including more labeled samples in estimating label distribution results in marginal, upper-bounded improvements.

Especially for CSS applications, model consumers are likely to care as much about estimating performance in their domain (to ensure validity) as they do about improving performance. An additional advantage of DsBias is that one can easily use two-fold estimation to effectively re-use any available labeled data for both estimating the label distribution and evaluating performance. That is, split the available labeled data in two, use half to estimate the label distribution, and the other half to estimate performance. Repeat this (reversing roles), and then take the average performance as an estimate of in-domain accuracy, without any model training or hyperparameter tuning required. One can then use all of the labeled data to estimate the label distribution for making predictions on the full unlabeled dataset. As shown in Figure 4, this produces an unbiased estimate, with variance that decreases with the amount of labeled data.

4.5 Domain Fine-tuning

One major advantage of contextual embedding models like RoBERTa is that one can easily fine-tune to a new domain by simply continuing to train on additional labeled data (Gururangan et al., 2020). Although this may not be a possibility for many model consumers (see §3.5), we evaluate this approach for the sake of completion.[9]

Here, we take the best-performing RoBERTa model from section §4.3, and fine-tune it with a

[9]Importantly, contextual embedding models can easily be *applied* with minimal computational requirements, but domain fine-tuning requires more resources and expertise.

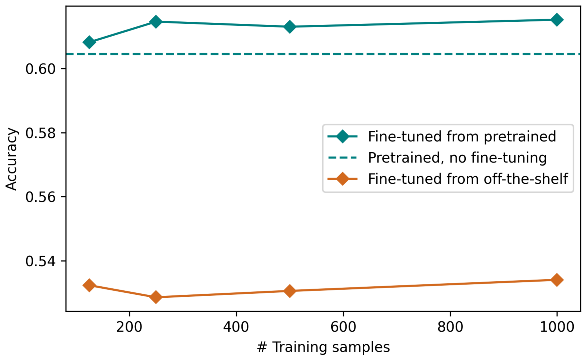

Figure 5: Mean validation accuracy on held-out domains of a RoBERTa+DsBias model on ARXIV, fine-tuned using a variable number of random samples from the heldout domain. In our experiments, fine-tuning a contextual embedding model pretrained for the same task on other domains is much better than simply fine-tuning an off-the-shelf model.

small number of samples from the unseen domain from the train split in the heldout domain, using a variable number of labeled samples, then evaluate the model using the validation split in the heldout domain. Figure 5 demonstrates that even with a relatively small number of labeled samples from the unseen domain, a second-pass fine-tuning results in a significant performance increase, but that the increase flattens out as number of samples increases. Of course, users will also need additional data for evaluating in-domain performance, so this underestimates the total amount of labeled data that would be required.

More importantly, we find that fine-tuning a model that has already been trained for the same task on out-of-domain data does far better than fine-tuning a generic off-the-shelf model, even with 1000 in-domain samples. Thus, despite the power of fine-tuning contextual embedding models, there is still a clear advantage for the CSS community of model producers creating such models for measuring categories of interest in text.

5 Discussion and Recommendations

A key idea of this paper is that domain adaptation should not be something that only model consumers have to confront. Rather, we should think of domain adaptation as a modular, collaborative process, in which model producers should anticipate that model consumer will want to apply models to new domains. Ideally, model producers would also make training data available to model consumers, so as to facilitate domain adaptation. For settings in which this is not possible, we have presented two techniques (DsBias and DsNorm) which improved performance for both logistic regression and contextual embedding models, and we encourage the development of additional techniques.

Although it is still useful for model producers to estimate and report model performance in the training domain(s) as part of model documentation (Mitchell et al., 2019), model consumers should not rely on such estimates when making use of off-the-shelf models. Rather, it is essential to have sufficient labeled data in the application domain so as to be able to estimate performance, in addition to any labeled data to be used for adaptation, and this should be budgeted for when planning annotations (Baheti et al., 2021). For specific applications, model consumers may also care about metrics beyond accuracy, and should evaluate models based on what is most relevant.

Lexicons such as LIWC have an enduring popularity, in part because of their ease of use. As the results above demonstrate, however, simple logistic regression models can do as well (in terms of classification accuracy). Contextual embedding models derived from the same data are considerably more accurate, and need not be any more difficult for practitioners to apply. Thus, we encourage CSS researchers to produce and share such models, even if the raw data itself cannot be shared.

6 Conclusion

Using off-the-shelf text classification models for computational social science requires careful thought regarding domain shift. In this paper, we propose to treat this as a modular process in which model producers can apply techniques of *anticipatory domain adaptation* to facilitate adaptation by model consumers. We demonstrate that using domain-specific bias (DsBias) and domain-specific normalization (DsNorm) produces a reliable performance boost for the model consumers, and that this applies to both linear and contextual embedding models. Finally, for cases where accuracy is more important than interpretability, we demonstrate the superior out-of-domain performance of contextual embedding models when compared to linear models, even without additional fine-tuning, and encourage model producers to make multiple types of models available.

References

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. 2019. Regularized learning for domain adaptation under label shifts. In *Proceedings of ICML*.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204.

Ashutosh Baheti, Maarten Sap, Alan Ritter, and Mark Riedl. 2021. Just say no: Analyzing the stance of neural dialogue generation in offensive contexts.

Dallas Card, Amber E. Boydstun, Justin H. Gross, Philip Resnik, and Noah A. Smith. 2015. The media frames corpus: Annotations of frames across issues. In *Proceedings of ACL*.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. In *Proceedings of EMNLP*.

Boris Chidlovskii, Stephane Clinchant, and Gabriela Csurka. 2016. Domain adaptation in the absence of source domain data. In *Proceedings of KDD*.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. 2019. On the use of arxiv as a dataset.

Johannes C. Eichstaedt, Robert J. Smith, Raina M. Merchant, Lyle H. Ungar, Patrick Crutchley, Daniel Preoţiuc-Pietro, David A. Asch, and H. Andrew Schwartz. 2018. Facebook language predicts depression in medical records. *Proceedings of the National Academy of Sciences*, 115(44):11203–11208.

Figure Eight. 2015. Twitter us airline sentiment. Data retrieved from Kaggle, `https://www.kaggle.com/crowdflower/twitter-airline-sentiment`.

Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. 2011. Sparse additive generative models of text. In *Proceedings of ICML*.

Jeremy A. Frimer, Reihane Boghrati, Jonathan Haidt, Jesse Graham, and Morteza Dehgani. 2019. Moral foundations dictionaries for linguistic analyses 2.0. Accessed: 2021-05-24.

Dean Fulgoni, Jordan Carpenter, Lyle Ungar, and Daniel Preoţiuc-Pietro. 2016. An empirical exploration of moral foundations theory in partisan news sources. In *Proceedings of LREC*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Justin Grimmer and Brandon M. Stewart. 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, 21(3):267–297.

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 ACL*.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of ACL*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168–177, New York, NY, USA. Association for Computing Machinery.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. 2007. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*.

Alison Huettner and Pero Subasic. 2000. In *ACL 2000 Companion Volume: Tutorial Abstracts and Demonstration Notes*.

C. J. Hutto and Eric Gilbert. 2014. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, 1.

Abigail Z. Jacobs and Hanna Wallach. 2021. Measurement and fairness. In *Proceedings of FAccT*.

Kokil Jaidka, Salvatore Giorgi, H. Andrew Schwartz, Margaret L. Kern, Lyle H. Ungar, and Johannes C. Eichstaedt. 2020. Estimating geographic subjective well-being from Twitter: A comparison of dictionary and data-driven language methods. *Proceedings of the National Academy of Sciences*, 117(19):10165–10171.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of ACL*.

Ananya Kumar, Tengyu Ma, and Percy Liang. 2020. Understanding self-training for gradual domain adaptation. In *Proceedings of ICML*.

Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. 2020a. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of CVPR*.

Sijia Li, Yilin Wang, Jia Xue, Nan Zhao, and Tingshao Zhu. 2020b. The impact of covid-19 epidemic declaration on psychological consequences: A study on active weibo users. *International Journal of Environmental Research and Public Health*, 17(6).

Jian Liang, Dapeng Hu, and Jiashi Feng. 2020. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of ICML*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

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.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of FAccT*.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. 29(3):436–465.

Dong Nguyen, Maria Liakata, Simon DeDeo, Jacob Eisenstein, David Mimno, Rebekah Tromble, and Jane Winters. 2020. How we do things with words: Analyzing text as social and cultural data. *Frontiers in Artificial Intelligence*, 3.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197, Hong Kong, China. Association for Computational Linguistics.

Reid Pryzant, Sugato Basu, and Kazoo Sone. 2018a. Interpretable neural architectures for attributing an ad's performance to its writing style. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 125–135, Brussels, Belgium. Association for Computational Linguistics.

Reid Pryzant, Dallas Card, Dan Jurafsky, Victor Veitch, and Dhanya Sridhar. 2021. Causal effects of linguistic properties. In *Proceedings of NAACL*.

Reid Pryzant, Kelly Shen, Dan Jurafsky, and Stefan Wagner. 2018b. Deconfounded lexicon induction for interpretable social science. In *Proceedings of NAACL*.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment.

Koustuv Saha, Benjamin Sugar, John Torous, Bruno Abrahao, Emre Kıcıman, and Munmun De Choudhury. 2019. A social media study on the effects of psychiatric medication use. In *Proceedings of the International AAAI Conference on Web and Social Media*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.

Yla R. Tausczik and James W. Pennebaker. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24—54.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of ACL*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's transformers: State-of-the-art natural language processing.

A Full Heldout Domain Accuracy

For each model-technique combination, for each dataset, and for each domain in the dataset, we train a model using the training split of all domains except the single heldout domain, then evaluate the model on the heldout domain, then average accuracy across these domains. These data were used to determine which model comparisons to test for significance, though we include all results on test data in the main paper for completeness.

		MFC		ARXIV		AMAZON		SENTI	
		acc	σ_Δ	acc	σ_Δ	acc	σ_Δ	acc	σ_Δ
LogReg	Base	0.501	-	0.541	-	0.672	-	0.647	-
	GR	0.502	0.002	0.542	0.003	0.709	0.001	0.638	0.003
	DSB	0.520	0.02	0.565	0.014	0.715	0.003	0.695	0.041
	DR	0.493	0.006	0.552	0.005	0.674	0.004	0.648	0.003
	DSN	0.452	0.013	0.483	0.033	0.682	0.012	0.595	0.044
	DSN+GR	0.453	0.013	0.483	0.033	0.681	0.012	0.595	0.044
	DSN+DSB	0.536	0.017	0.570	0.013	0.717	0.002	0.712	0.039
	DSN+DR	0.451	0.015	0.358	0.035	0.491	0.016	0.609	0.044
RoBERTa	Base	0.581	-	0.583	-	0.772	-	0.803	-
	DR	0.585	0.014	0.587	0.005	0.782	0.017	0.817	0.012
	GR	0.204	0.046	0.510	0.01	0.778	0.012	0.684	0.068
	DSB	0.615	0.031	0.605	0.011	0.779	0.012	0.819	0.014

Table 3: Validation accuracy of models trained holding out one domain per trial, then evaluated on the heldout domain, for all configurations of each model. σ_Δ is the standard deviation of accuracy difference in each domain over the corresponding baseline ("Base").

B Comparison to Off-the-shelf Sentiment Models and Lexicons

To ensure that our linear model achieve reasonable performance, we compare our best logistic regression classifiers (using DSN+DSB) to several off-the-shelf sentiment lexicons and models, applied to the SENTI dataset. For each lexicon, we use the available (weighted or unweighted) word list as features, and introduce a learnable threshold, which we fine tune to each target domain in turn, using the same 250 samples from that domain as we use to estimate label distribution for our best model.

Results are shown in Table 4. Notably, not only does performance vary across lexicons (showing the sensitivity of results to which lexicon is chosen), but none do as well as our best linear mode, indicating that even commercial packages such as LIWC (Tausczik and Pennebaker, 2010) are no better at generalizing to new domains that a regularized logistic regression model.

Model / Lexicon	Untuned Acc	Tuned Acc (250 samples)
VADER (Hutto and Gilbert, 2014)	0.631	-
General Inquirer (Stone et al., 1966)	0.635	0.675
SentiWordNet (Baccianella et al., 2010)	0.608	0.680
LIWC (Tausczik and Pennebaker, 2010)	0.648	0.689
Opinion Lexicon (Hu and Liu, 2004)	0.680	0.706
LogReg	0.647	0.712

Table 4: Validation accuracy in unseen domains of popular off-the-shelf sentiment lexicons in comparison to our best model. For LogReg, "untuned" refers to its baseline, and "tuned" is the model with DSN and DSB applied with estimated label distribution. VADER is not tuned as it is distributed as a classifier [DC: dbl check].

C Single Domain Training

Similar to the previous experiment where we held out a single domain, here we train only on a single domain, and evaluate with all non-training domains.

		MFC		ARXIV		AMAZON		SENTI	
		acc	σ_Δ	acc	σ_Δ	acc	σ_Δ	acc	σ_Δ
LogReg	Base	0.426	-	0.555	-	0.653	-	0.574	-
	GR	0.425	0.0	0.554	0.0	0.652	0.001	0.572	0.002
	DSB	0.447	0.006	0.596	0.008	0.681	0.016	0.670	0.018
	DR	0.423	0.002	0.574	0.012	0.605	0.002	0.571	0.006
	DSN	0.366	0.01	0.417	0.019	0.629	0.015	0.545	0.013
	DSN+GR	0.366	0.012	0.415	0.02	0.629	0.015	0.545	0.013
	DSN+DSB	0.472	0.008	0.598	0.007	0.683	0.015	0.670	0.018
	DSN+DR	0.378	0.005	0.349	0.018	0.481	0.025	0.549	0.015
RoBERTa	Base	0.48	-	0.539	-	0.727	-	0.622	-
	DR	0.510	0.023	0.542	0.004	0.736	0.028	0.620	0.014
	GR	0.168	0.034	0.448	0.074	0.647	0.026	0.548	0.062
	DSB	0.540	0.029	0.560	0.008	0.751	0.023	0.699	0.039

Table 5: Validation accuracy of models trained with a single domain, then evaluated on all other domains combined, for all configurations of each model. σ_Δ is the standard deviation of accuracy difference in each domain over the corresponding baseline (Base).

In single domain training, since no deconfounding between training domain is possible, gradient reversal (GR) and deep residualization (DR) fails to meaningfully improve performance.

Comparing table 5 to table 3, not only do we observe a very similar trend of performance differences, where our recommended model-technique combinations (Lexicon+DSN+DSB, RoBERTa+DSB) consistently outperforms the rest, but the difference is more pronounced.

D Out-of-domain Performance Drop

	MFC			ARXIV			AMAZON			SENTI		
	ID	OOD	σ_Δ	ID	OOD	σ_Δ	ID	OOD	σ_Δ	ID	OOD	σ_Δ
LogReg	0.607	0.508	0.036	0.583	0.542	0.012	0.722	0.672	0.062	0.756	0.649	0.060
RoBERTa	0.703	0.600	0.071	0.608	0.571	0.021	0.797	0.772	0.021	0.837	0.789	0.073

Table 6: Test accuracy of models trained on all domains then evaluated on the test split of each domain (in-domain “ID"), and trained on all but one held-out domain then evaluated on the test split of that held-out domain (out-of-domain “OOD”). σ_Δ is the standard deviation of accuracy difference in each domain.

E Estimating Performance

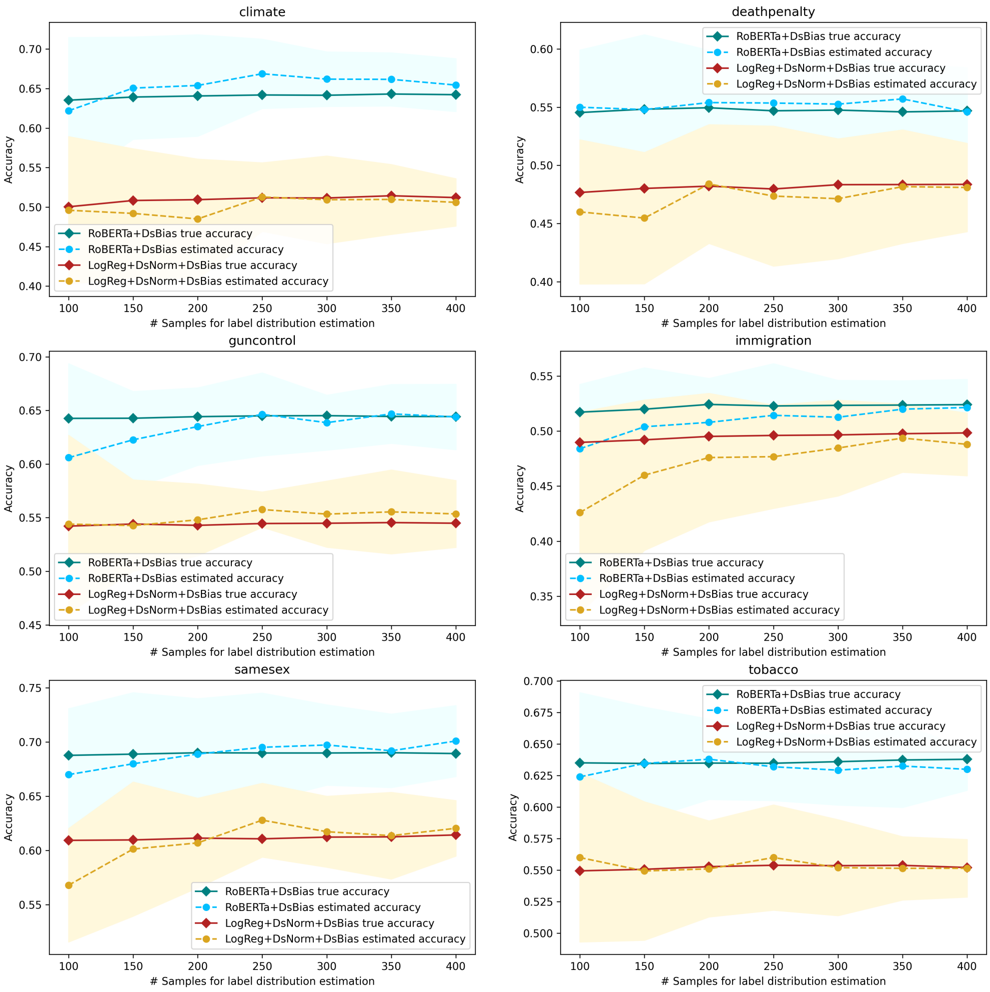

Figure 6: Validation accuracy calculated from all holdout samples, and from limited samples, of each topic (domain) in the Media Frame Corpus (MFC). Shaded area denotes 1 standard deviation from mean estimated performance

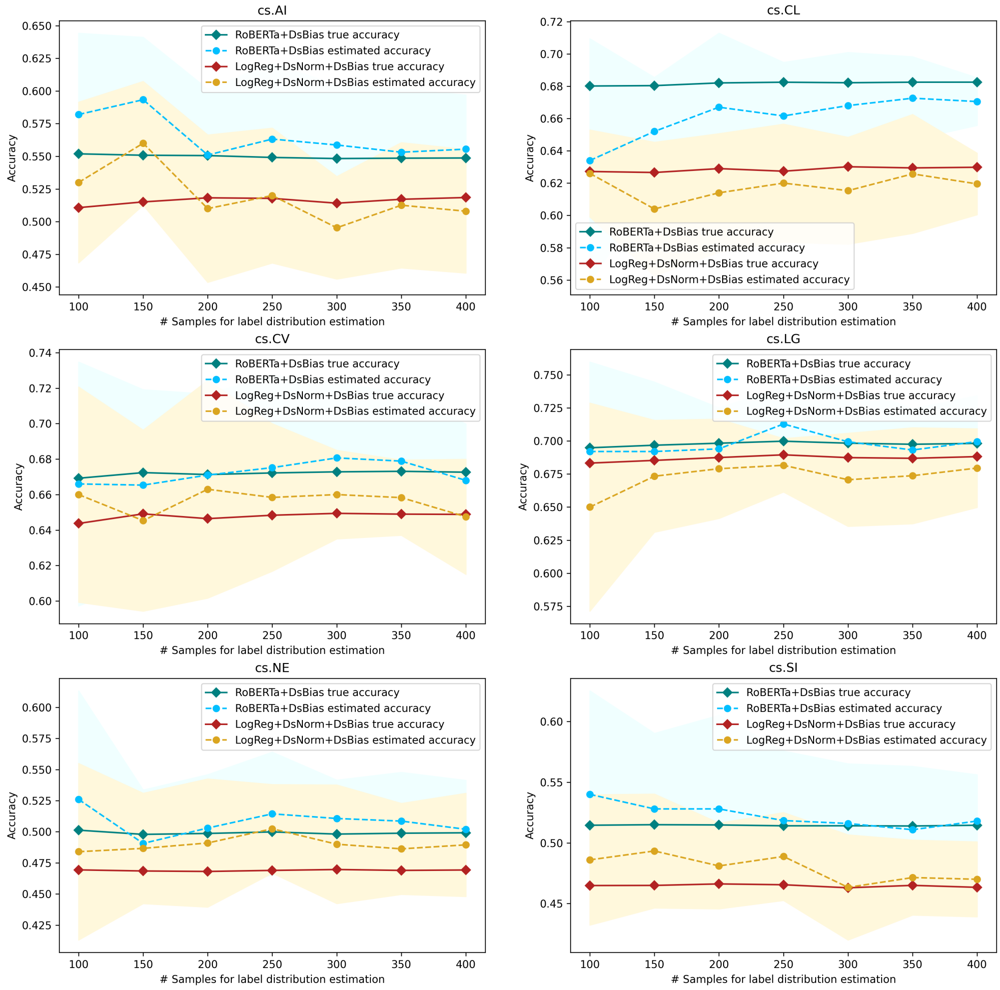

Figure 7: Validation accuracy calculated from all holdout samples, and from limited samples, of each category (domain) in ARXIV. Shaded area denotes 1 standard deviation from mean estimated performance

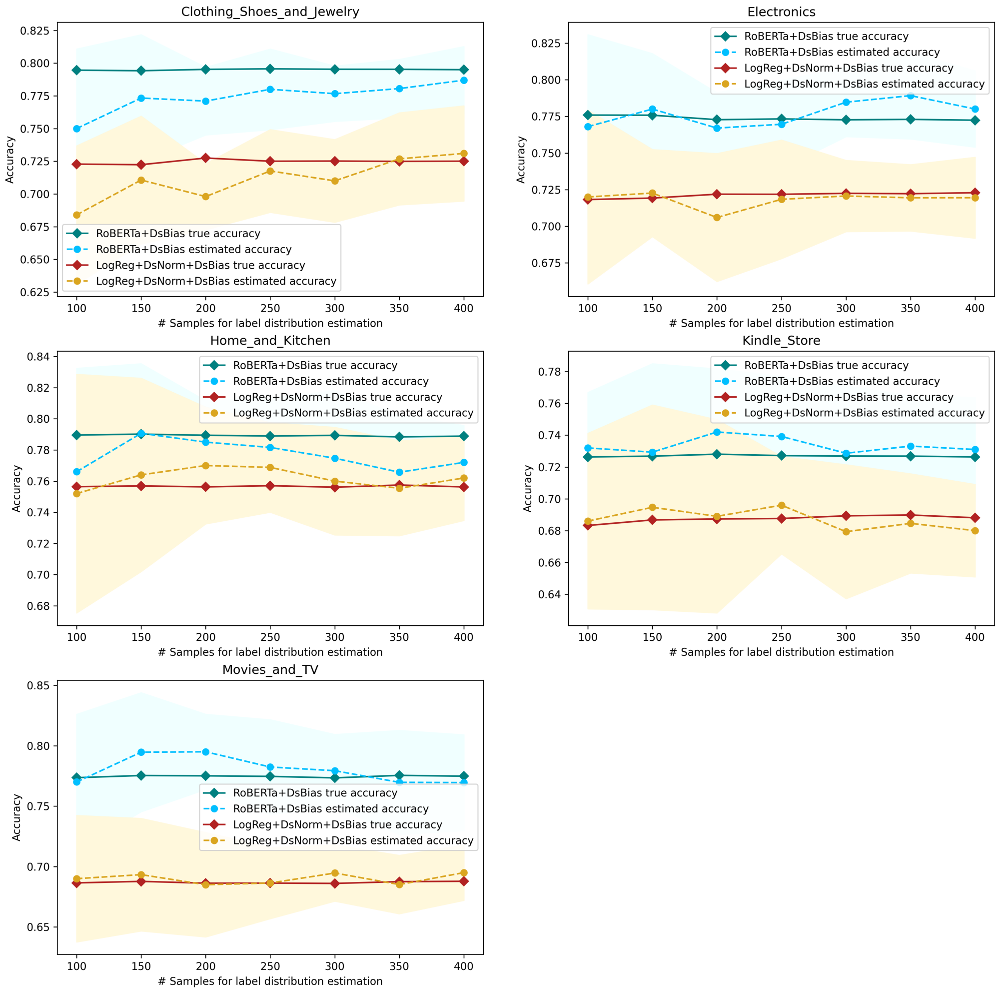

Figure 8: Validation accuracy calculated from all holdout samples, and from limited samples, of each category (domain) in AMAZON. Shaded area denotes 1 standard deviation from mean estimated performance

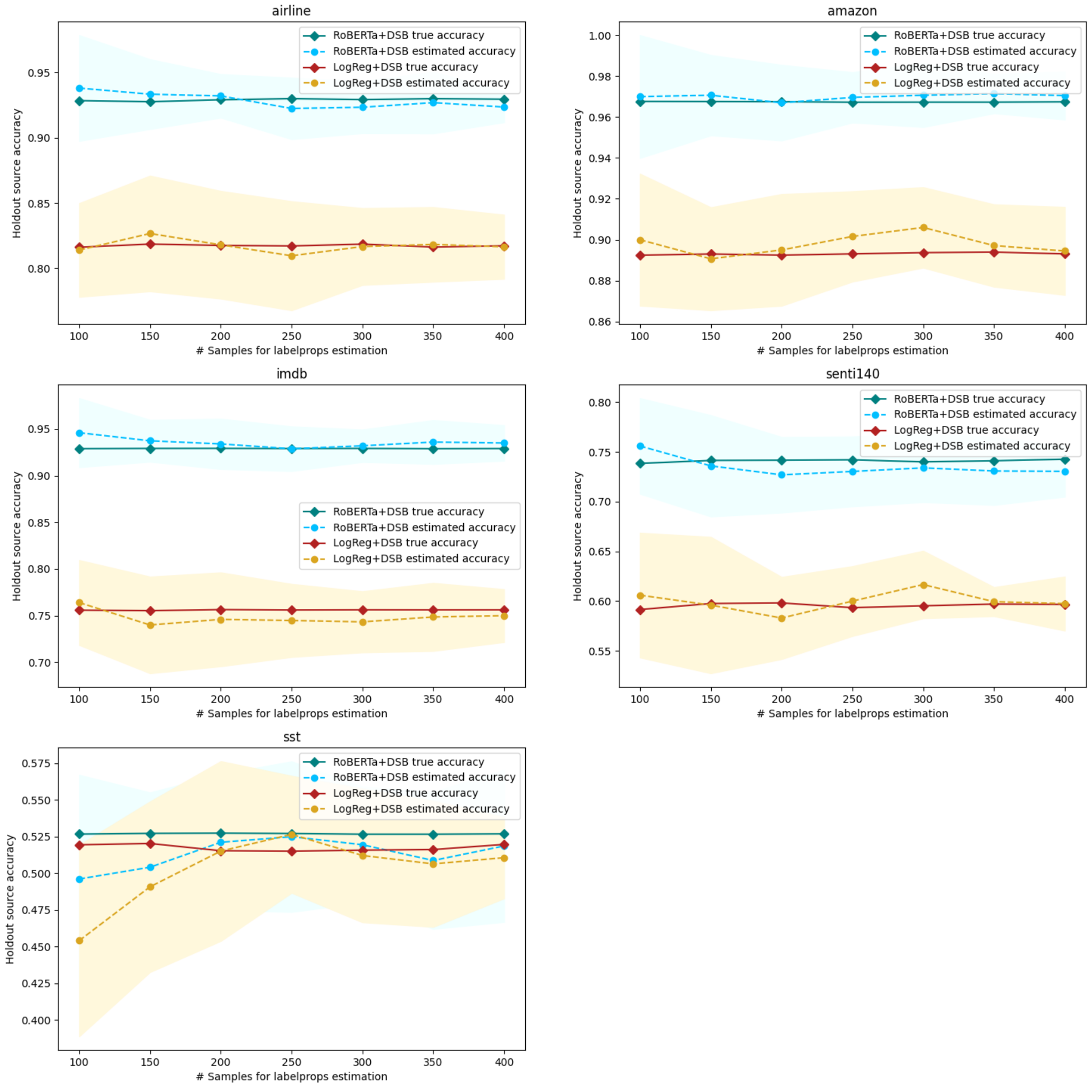

Figure 9: Validation accuracy calculated from all holdout samples, and from limited samples, of each sub-dataset (domain) in SENTI. Shaded area denotes 1 standard deviation from mean estimated performance

F Example Lexicon

Economic	Capacity and Resources	Morality	Fairness and Equality	Legality, Constitutionality, Jurisdiction	Policy Prescription and Evaluation	Crime and Punishment	Security and Defense
economic	applications	moral	discrimination	asylum	ordinance	criminals	terrorist
financial	shortage	church	fairness	lawsuit	rid	deport	security
budget	species	pope	black	justices	punishment	deported	terrorists
business	capacity	catholic	equality	sued	vehicles	allegedly	border
economy	ocean	churches	innocent	suing	policy	injection	military
fund	handle	leaders	race	constitution	penalty	minors	patrol
jobs	process	christian	racial	plaintiffs	citizenship	smuggling	fbi
costs	surge	religious	equal	lawsuits	effect	kill	terror
economists	science	rev	innocence	visa	plan	crackdown	threats
sales	resources	francis	evidence	suit	bill	deportation	pentagon
corporate	scientists	bishop	unfair	court	ban	fine	intelligence
company	foreign	faith	fair	visas	would	police	terrorism
companies	wait	rabbi	blacks	judge	policies	investigators	protect
tax	critical	churchs	testimony	attorney	smokefree	firstdegree	guard
cost	waiting	jewish	facts	antonin	proposal	prison	war
revenue	years	society	civil	militia	bans	maximum	secure
stores	tons	clergy	racist	shall	supporters	arrested	airports
treasury	growing	christians	true	lawyers	designated	sentenced	attacks
dollars	used	nicotine	equally	licenses	buildings	scheme	russian
money	lines	bible	treated	granted	homeland	executed	defense

Health and Safety	Quality of Life	Cultural Identity	Public Sentiment	Political	External Regulation and Reputation	Other
mentally	daughter	documentary	poll	governor	countries	hillary
health	loved	film	protesters	republicans	minister	chris
condition	benefits	movie	rally	bloombergs	mexican	gop
medical	quit	culture	protest	conservatives	foreign	annual
disease	mother	actor	marched	sen	european	paid
doctors	weather	cultural	demonstrators	clinton	un	brother
suicide	college	book	voters	reelection	mexicans	cultural
hospital	families	ethnic	activists	bipartisan	visit	money
pain	tears	executions	organizers	gop	france	supporting
safe	temperatures	population	organized	mayor	states	stores
safety	felt	english	gathered	hillary	china	accused
mental	family	movies	protests	statements	negotiations	interests
lung	everything	history	mom	rep	agreement	governors
coverage	temperature	players	polls	cuomo	united	candidate
locks	living	tv	polling	mayors	talks	fund
retarded	married	census	mothers	endorsement	mexico	endorsement
lungs	conditions	league	attitudes	obama	summit	didnt
risk	life	decline	nra	referendum	australia	economic
illness	classes	star	signatures	ryan	mexicos	reelection
diseases	father	smoked	organization	republican	canadian	shortly

Table 7: Top weighted 20 words from each class in a lexicon elicited from the Media Frame Corpus (MFC), with a logistic regression model and using Domain-Specific Bias (DSB) and Domain-Specific Normalization (DSN). Weight value associated with each word not included.

-2008	2009-2014	2015-2018	2019-
rules	web	recurrent	covid19
grammar	bayesian	deep	bert
presented	belief	convolutional	federated
logic	variables	neural	transformer
described	markov	lstm	selfsupervised
grammars	graphical	big	fewshot
theory	svm	adversarial	pandemic
statistical	technique	pascal	transformerbased
describes	probabilistic	endtoend	fairness
parsing	words	embeddings	selfattention
information	propagation	reinforcement	sota
linguistic	probabilities	nonconvex	transformers
general	convex	stateoftheart	ai
syntactic	recognition	dataset	explainable
disambiguation	svms	propose	downstream
shown	database	sentiment	explainability
sense	independence	convnet	outofdistribution
definition	conditional	stochastic	nas
discussed	uncertainty	mnist	learningbased
tested	basis	dropout	embeddings
class	immune	atari	code
notion	em	rnn	backbone
semantics	sparse	sequencetosequence	gnns
presents	dictionary	generative	gnn
programming	wavelet	train	augmentation
programs	sound	gradient	quantum
order	collaborative	embedding	continual
algorithm	extraction	convnets	lightweight
classes	management	explore	neural
two	coding	machine	unet
noun	techniques	jointly	module

Table 8: Top weighted 30 words from each class in a lexicon elicited from the abstract texts in the arXiv dataset (ARXIV), with a logistic regression model and using Domain-Specific Bias (DSB) and Domain-Specific Normalization (DSN). Weight value associated with each word not included.

Negative (1 star)	Neutral (2-4 stars)	Positive (5 stars)
waste	ok	love
poor	stars	perfect
junk	okay	excellent
horrible	however	awesome
terrible	disappointing	loves
worst	otherwise	perfectly
awful	unfortunately	great
return	complaint	highly
returned	overall	glad
cheaply	downside	loved
useless	returned	amazing
boring	bit	pleased
poorly	reason	beautiful
broke	cute	thank
garbage	returning	wonderful
disappointed	little	thanks
nothing	wish	happy
disappointing	though	fantastic
died	good	favorite
apart	slow	comfortable
cheap	decent	compliments
crap	flimsy	wait
defective	annoying	gorgeous
refund	stiff	exactly
returning	runs	best
money	issue	worried
month	liked	admit
beware	missing	happier
uncomfortable	interesting	wow
fell	nice	worry
stopped	alright	adorable
star	overpriced	faster
disappointment	except	nice
completely	problem	helps
weak	expected	incredible
description	awkward	classic
even	gave	satisfied
bad	thinner	originally
within	flaw	charm
minutes	cons	classy
broken	concept	durable
cannot	sometimes	needed
shame	seems	fast
worse	mechanism	comfy
unless	bulky	beautifully
piece	lack	truly
barely	pretty	recently
stuck	narrow	easier
ripped	meh	ram
please	careful	cleans

Table 9: Top weighted 50 words from each class in a lexicon elicited from amazon review texts (AMAZON), with a logistic regression model and using Domain-Specific Bias (DSB) and Domain-Specific Normalization (DSN). Weight value associated with each word not included.

Negative	Positive
poorly	thank
annoying	thanks
worst	superb
boring	hi
hurts	amazing
waste	brilliant
dislike	excellent
ugh	subtle
finale	smooth
disappointed	awesome
sad	wonderfully
poor	outstanding
wooden	hahaha
redeeming	yay
cancelled	excited
sucks	hilarious
wanna	notice
disappointment	seemingly
bag	funniest
unfortunately	safe
ugly	noir
mediocre	impressed
laughable	extraordinary
crappy	haha
lousy	powerful
turkey	humorous
claims	loved
sorry	solid
junk	helpful
arms	higher
sick	germany
awful	dvd
disappointing	ideal
pointless	sweet
shots	twenty
barely	great
confused	pleasure
headache	friday
ruined	happy
ticket	independent
potential	involve
obnoxious	masterpiece
luggage	captures
shallow	welcome
pain	rare
anymore	cool
nowhere	south
terrible	incredible
miss	best
min	gripping

Table 10: Top weighted 50 words from each class in a lexicon elicited from a collection of multiple sentiment classification datasets (SENTI), with a logistic regression model and using Domain-Specific Bias (DSB) and Domain-Specific Normalization (DSN). Weight value associated with each word not included.

G Data Splits

For the Media Frame Corpus (MFC), we a fixed number of 400 random samples from each news issue (domain) as the test set, and do not use them for any training or hyperparameter tuning until the end for reporting test performance. Validation data for hyperparameter tuning in experiments is either from a held-out source, or k-fold validation.

	Climate	Gun control	Death penalty	Immigration	Same-sex marriage	Tobacco	Total
Train	3795	3777	8498	5533	3956	3251	28810
Test	400	400	400	400	400	400	2400
Total	4195	4177	8898	5933	4356	3651	31210

Table 11: Sample sizes of each domain and each split from the Media Frame Corpus (MFC)

For the arXiv dataset (ARXIV), we take a fixed proportion of 10% of random samples from each paper category (domain) as the test set, and do not use them for any training or hyperparameter tuning until the end for reporting test performance. Validation data for hyperparameter tuning in experiments is either from a held-out source, or k-fold validation.

	Artificial intelligence (cs.AI)	Computation and language (cs.CL)	Computer vision (cs.CV)	Machine learning (cs.LG)	Neural and evolutionary computing (cs.NE)	Social and Information Networks (cs.SI)	Total
Train	18294	21131	46008	53647	4798	11086	154986
Test	2034	2350	5113	5962	534	1233	17226
Total	20328	23481	51121	59609	5332	12319	172212

Table 12: Sample sizes of each domain and each split from the arXiv dataset (ARXIV)

For the Amazon reviews dataset AMAZON, we first subsample to keep only 0.2% of the original dataset size to simulate a data-scarce setting. We then take a fixed proportion of 10% of random samples from each category (domain) as the test set, and do not use them for any training or hyperparameter tuning until the end for reporting test performance. Validation data for hyperparameter tuning in experiments is either from a held-out source, or k-fold validation.

	Clothing, Shoes and Jewelry	Electronics	Home and Kitchen	Kindle Store	Movies and TV	Total
Train	20315	12132	12418	4002	6140	55007
Test	2258	1350	1382	446	683	6119
Total	22573	13482	13800	4448	6823	61126

Table 13: Sample sizes of each domain and each split from the Amazon review dataset (AMAZON)

For SENTI, we take a fixed proportion of 10% of random samples from each data source (domain) as the test set, and do not use them for any training or hyperparameter tuning until the end for reporting test performance. Validation data for hyperparameter tuning in experiments is either from a held-out source, or k-fold validation.

	Airline Tweets	Amazon Books	IMDb Movie Reviews	Sentiment 140	Stanford Sentiment Treebank	Total
Train	7080	7843	8977	9002	2778	35680
Test	788	873	999	1001	310	3971
Total	7868	8716	9976	10003	3088	39651

Table 14: Sample sizes of each domain and each split from the sentiment classification dataset collection (SENTI)

H Data Preprocessing

Sample texts are preprocessed before used to train models and perform experiments. For both types of models, urls are first removed from the text. If the text is from a Tweet, then Twitter handlers (tokens starting with @) and emojis are also identified and removed.

For RoBERTa models, this sanitized text is then passed into a tokenized as-is without any additional processing. For logistic regression models, we then build a bag-of-word feature vector by first removing all punctuation, special symbols, English stopwords, pure numbers, and tokens including both alphabetical and numeric characters. Finally, we build a vocabulary of a fixed size of 5000 most frequent tokens, and convert the preprocessed texts into feature vectors.

I Experiment Setup and Hyperparameter Tuning

As in section §4.3 and section §4.5 we train multiple models of various configurations using different combination of training domains, we maintain a consistent strategy for hyperparameter tuning to ensure performance comparability.

Logistic regression models have one hyperparameter, the L1 regularization constant λ. For each experiment and each model configuration, we first run k-fold validation within the train set, and conduct a search for $\lambda = 1^{-5} \times 2^k, k \in (0, 4)$, while optimizing for lowest loss on the main prediction target on the validation set. Then we use the same optimal λ to train with the full train set until convergence.

RoBERTa models have one hyperparameter, the number of epochs E to train or fine-tune. Since deep contextual embedding models are very powerful in the context of our small datasets, we early-stop during training to ensure it does not overfit to the training data. For each experiment and each model configuration, we first run k-fold validation within the train set, and conduct a search for $E \in (1, 8)$ for the out-of-domain experiments, and for $E \in (1, 15)$ the domain fine-tuning experiments, while optimizing for lowest loss on the main prediction target on the validation set. Then we use the full train set and train for the same optimal E epochs.

J Power Analysis

Model A Model B	LogReg LogReg+DSB+DSN		LogReg+DSB LogReg+DSB+DSN		RoBERTa RoBERTa+DSB	
	McNemar's p	Power	McNemar's p	Power	McNemar's p	Power
MFC	2.18e-06	1.0	0.0327	0.362	0.009	0.908
ARXIV	1.66e-24	1.0	0.0055	0.281	9.62e-11	1.0
AMAZON	0.0039	0.491	0.0787	0.414	3.58e-06	0.952
SENTI	6.52e-18	1.0	5.13e-05	0.968	0.0002	0.934

Table 15: Power analysis values for pitting different configurations of interest against each other. McNemar's p is calculated using the test split. Statistical power is calculated per Card et al. (2020) using all validation samples, with dataset size equivalent to that of the test split.