To Annotate or Not?
Predicting Performance Drop under Domain Shift

Hady Elsahar and **Matthias Gallé**
NAVER LABS Europe
{hady.elsahar,matthias.galle}@naverlabs.com

Abstract

Performance drop due to domain-shift is an endemic problem for NLP models in production. This problem creates an urge to continuously annotate evaluation datasets to measure the expected drop in the model performance which can be prohibitively expensive and slow. In this paper, we study the problem of predicting the performance drop of modern NLP models under domain-shift, in the absence of any target domain labels. We investigate three families of methods ($\mathcal{H}$-divergence, reverse classification accuracy and confidence measures), show how they can be used to predict the performance drop and study their robustness to adversarial domain-shifts. Our results on sentiment classification and sequence labelling show that our method is able to predict performance drops with an error rate as low as 2.15% and 0.89% for sentiment analysis and POS tagging respectively.

1 Introduction

Building Natural Language Processing models that perform well in the wild is still an open and challenging problem. It is well known that modern machine-learning models can be brittle, meaning that – even when achieving impressive performance on the evaluation set – their performance can degrade significantly when exposed to new examples with differences in vocabulary and writing style (Blitzer and Pereira, 2007; Jia and Liang, 2017; Brun and Nikoulina, 2018). This drop in performance when changing from domain D_s to domain D_t can be due to a variety of causes. It could be because of *Co-variate Shift* (Shimodaira, 2000; Storkey, 2009), where the input distribution changes, but the conditional distribution does not, i.e. $P_{D_s}(y|x) = P_{D_t}(y|x)$ but $P_{D_s}(x) \neq P_{D_t}(x)$; or due to *Concept Shift*, when $P_{D_s}(y|x) \neq P_{D_t}(y|x)$ and $P_{D_s}(x) = P_{D_t}(x)$, or *Label Shift*, when $P_{D_s}(y) \neq P_{D_t}(y)$ and $P_{D_s}(x|y) = P_{D_t}(x|y)$ (Zhang et al., 2013; Lipton et al., 2018), or a mix between them (Moreno-Torres et al., 2012; Quionero-Candela et al., 2009).

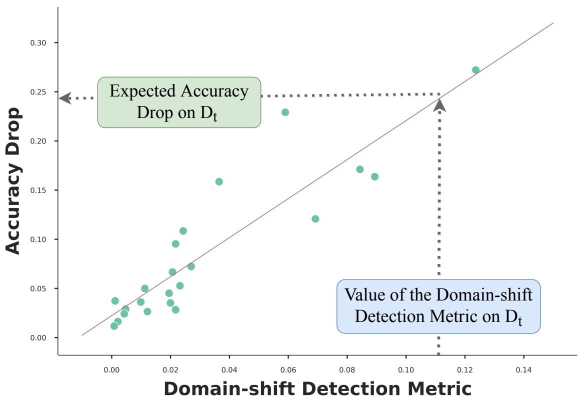


Figure 1: In this paper we introduce several domain-shift detection metrics (x-axis) and employ them to estimate the performance drop on a new target domain D_t by regressing on those metrics and their associated real performance drop (green dots).

Such changes are often the norm in many real-world applications, creating an urge in the industry to continuously check that models in production are performing well in the wild or when there is a new client or a new type of data. However, continuously sampling and annotating data-points can be prohibitively slow and costly, particularly when a large annotated sample is needed to correctly represent the target joint distribution or when the change can be gradual (Kifer et al., 2004).

In this work, we investigate how we can estimate the performance drop of a model when evaluated on a new target domain, without the need of any labeled examples from this target domain. Performing this estimation accurately has an important impact on the decision process of real-time debugging and maintaining machine learning models in production. For instance, such insights can drive

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing, pages 2163–2173,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

the decision to annotate more data for retraining or even adjusting the model accordingly (e.g. performing unsupervised domain adaptation).

We propose a method that takes advantage of several domain-shift detection metrics and employ them to estimate through regression the performance drop on a target domain on which no annotated data is available, The overall approach is schematically depicted in Fig. 1. The relationship between the domain shift metrics and the real performance drop of different domains is precomputed over different existing domains (green dots in Fig. 1), and for a new target domains the performance drop can be estimated simply by evaluating the function learned through simple regression. This process directly yields the value of performance drop the model will suffer when exposed to this target domain examples. We believe is more interpretable and useful to ML practitioners than other intermediate signals such as of out-of-distribution detection methods.

This paper introduces the following contributions:

- We introduce a new task and methodology for directly predicting performance drop of a model under domain-shift, without the need of labeled examples from the target domain.
- We survey, formalize and evaluate domain-shift detection metrics from 3 different families (§2) and propose new adaptations.
- We benchmark each proposes metric on two tasks of different natures: document classification and sequence labeling (§3 and §4), and show their robustness under adversarial domain-shift scenarios.

2 Measuring Performance Drop

We are interested in the problem of measuring the performance drop of classifier $\mathcal{C}$, trained on samples from the source domain D_s when applied to the target domain D_t samples. In the presence of labeled samples from D_t, this could empirically be measured by the difference in test errors between the source and target domain.

$$\Delta\mathcal{R} = \underset{(x,y)\in D_s}{\boldsymbol{Pr}}(\mathcal{C}(x) \neq y) - \underset{(x,y)\in D_t}{\boldsymbol{Pr}}(\mathcal{C}(x) \neq y) \quad (1)$$

To calculate this value empirically, one would need annotated examples in the form of a labeled test set for each new target domain, which is costly and/or time-consuming. In this section we introduce several metrics of different natures that correlate with the drop in a model performance without the need of any annotated examples from the target domain. In particular, we will consider three families of measures:

- **$\mathcal{H}$-divergence based metrics**: based on the capacity of another classification model to distinguish between samples from D_s and D_t.
- **Confidence-based**, using the certainty of the model over its predictions.
- **Reverse classification accuracy**, where predicted values are used as pseudo-labels over D_t.

2.1 $\mathcal{H}$-divergence based metrics

Building upon previous work from Kifer et al. (2004) for detecting domain change in data streams, Ben-David et al. (2010) define the target error of a model under domain-shift in terms of its source error and the divergence between the distributions of the two domains. This domain divergence is referred to as the $\mathcal{H}$-divergence and formalized as follows: given a hypothesis class $\mathcal{H}$ that consists of a set of binary classifiers $h : \mathcal{X} \rightarrow \{0, 1\}$ the $\mathcal{H}$-divergence can be represented as:

$$2 \sup_{h\in H} |Pr_{x\sim D_s}[h(x) = 1] - Pr_{x\sim D_t}[h(x) = 1]| \quad (2)$$

This translates to calculating the capacity of the hypothesis class $\mathcal{H}$ to distinguish between the distributions of both domains D_s and D_t.[1] Ben-David et al. (2010) prove that for a symmetric hypothesis class the $\mathcal{H}$-divergence can be calculated through a finite sample sampled from D_s and D_t.[2] Calculating the value for $\mathcal{H}$-divergence exactly requires finding the hypothesis $h \in \mathcal{H}$ that has minimum error on the binary classification problem between samples from D_s and D_t, which is intractable in practice. Thus, Ben-David et al. (2006) approximates this through learning a model that discriminates between the source and target examples. This approximated value is often referred to as the Proxy $\mathcal{A}$-distance (**`PAD`**).
Given a domain classifier $G_d : x \rightarrow [0, 1]$ parameterized by θ_d, we calculate **`PAD`** as the following:

[1] For simplicity we abuse the notation here by using D_s and D_t to refer both to the distribution and the set of samples of the source and target domain respectively.

[2] See (Ben-David et al., 2010, Appendix) for the proof.

$$
\begin{gathered}
\texttt{PAD} = 1 - 2\mathcal{E}(G_d) \\
s.t.; \\
\mathcal{E}(G) = 1 - \frac{1}{|D|} \sum_{x_i \in D_s, D_t} |G(x_i) - \mathbb{I}(x_i \in D_s)|
\end{gathered} \tag{3}
$$

where $\mathbb{I}$ is an indicator function.

The **PAD** measure is task-agnostic and measures therefore only the co-variate shift. It has been used before to measure domain discrepancy between datasets (Blitzer et al., 2007; Rai et al., 2010) for NLP applications. However, different from the time when **PAD** was introduced, modern NLP models not only compute a mapping between input and labels, but also infer an intermediate representation. For a given task, this representation is supposed to provide a view of the input that highlights the relevant part that could be helpful for a correct classification in this task. In particular, those intermediate representations should not be sensitive to task-irrelevant features that provide nevertheless strong signals to distinguish between the source and the target domains (yielding high **PAD** values). As an example, consider the task of named entity extraction on top of the 20 newsgroup dataset: the `To:` field is a highly discriminating feature for domain classification but arguably irrelevant to the task of extracting named entities.

Following this intuition, we propose a modification to the **PAD** measure. It is the classification accuracy of discriminating between the intermediate representation coming from – respectively – source and target domain (we use the last layer of the neural network). We assume that the task classifier $\mathcal{C}$ consists of two functions G_f and G_y. The first projects the input to a hidden representation of size m: $G_f : X \rightarrow \mathbb{R}^m$; while the second is a linear layer that uses this representation to predict the class labels $G_y : \mathbb{R}^m \rightarrow [0,1]^{|Y|}$. Differently from **PAD**, the domain classifier $G_d^* : \mathbb{R}^m \rightarrow [0,1]$ takes the hidden representations as an input instead of the original input. The learnable parameters of G_f, G_y and G_d^* are θ_f, θ_y and θ_d^* respectively. Our proposed metric **PAD⋆** is then:

$$
\texttt{PAD}\star = 1 - 2\mathcal{E}\left(G_d^*(G_f(x)\right)
$$

θ_f, θ_y are learned by minimizing the loss function of the task. Afterwards θ_f is frozen and θ_d^* is learned by minimizing the negative log-likelihood loss[3] for the domain discrimination task between D_s and D_t.

2.2 Confidence Based Metrics

While the final decision of classifiers is discrete, the weight given to that decision can be interpreted as the confidence the model has in that decision. This has been the basis of overcoming domain-shift using many self-training techniques which select the most confident examples as new training examples, together with the predicted class as pseudo-label (McClosky et al., 2006). However, modern neural networks are known to give wrongly calibrated confidence scores (Guo et al., 2017) meaning that the associated probability scores to the predicted class label does not reflect its correctness likelihood.

A few calibration techniques have been proposed to overcome this problem. For its simplicity and effectiveness we follow the temperature scaling method (Guo et al., 2017). It is a post training method that rescales the logits of any neural network model to soften the softmax by raising the output entropy of the probabilities scores.

Given a model trained on the source domain dataset D_s, let z be the logits vector produced by the very last layer for a given input, yielding the non-calibrated confidence score $q = \max_k(\mathit{softmax}(z))^k$ for the predicted class label. The calibrated confidence score $\widehat{q}$ is calculated then as follows:

$$
\widehat{q} = \max_k(\mathit{softmax}(z/T))^k \tag{4}
$$

Where T is a learnable scalar "temperature" parameter. T can be learned by minimizing the negative log likelihood loss over the validation set D_s^{val};[4]

$$
T^* = \underset{T}{\boldsymbol{argmin}} \left(-\sum_{i=1}^{N} \mathbb{1}_{[k=y_i]} \, log\left(\mathit{softmax}\left(z_i/T\right)\right)\right) \\
(x_i, y_i) \in D_s^{val}, \; T > 0 \tag{5}
$$

Where $\mathbb{1}_{[k=y_i]}$ is a one hot vector containing one in front of the true class label y_i and zero elsewhere.

Accordingly, we introduce two confidence based metrics to measure the domain-shift between the

[3] Minimizing the Huber loss as in (Ben-David et al., 2006) provides very similar results.

[4] Following (Guo et al., 2017) we use for this validation set the same set as for hyper-parameter tuning.

source and the target datasets D_s and D_t.

1) **CONF** the drop in average probability scores of the predicted class:

$$\textbf{CONF} = \frac{1}{|D_s|} \sum_{i:x_i \in D_s} q_i - \frac{1}{|D_t|} \sum_{j:x_j \in D_t} q_j \quad (6)$$

2) **CONF_CALIB** the drop in average calibrated probability scores for the predicted class:

$$\textbf{CONF\_CALIB} = \frac{1}{|D_s|} \sum_{i:x_i \in D_s} \widehat{q_i} - \frac{1}{|D_t|} \sum_{j:x_j \in D_t} \widehat{q_j} \quad (7)$$

2.3 Reverse Classification Accuracy

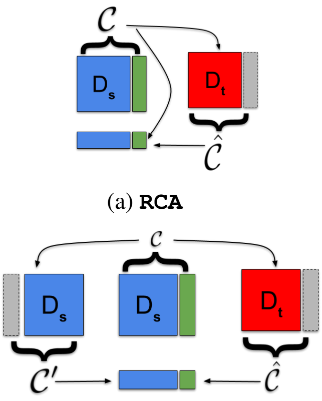
The idea of reverse classification accuracy is to use a classifier trained on the source domain to pseudo-label the target domain. That new dataset is then used to train a new classifier whose accuracy is measured on held-out data from the source domain. This was used in the past to select among existing models or datasets the best-performing one for a given target domain (Fan and Davidson, 2006; Zhong et al., 2010). In order to use this to create a proxy for domain-shift, we proceed as follows: a task classifier $\mathcal{C}$ is trained on the annotated source domain dataset D_s which is then run to create pseudo-labels for the unlabeled target data D_t. Those pseudo-labels are then used as training data for a reverse classifier $\hat{\mathcal{C}}$ – using the same architecture and training algorithm used to obtain $\mathcal{C}$. The performance of both classifiers are compared on a heldout subset $D_s^{'} \in D_s$ from the source domain datasets and used to define the **RCA** measure

$$\textbf{RCA} = \frac{\mathit{1}}{|D_s^{'}|} \sum_{x_i, y_i \in D_s^{'}}^{m^{'}} \mathbb{I}\left[y_i = \mathcal{C}(x_i)\right] - \mathbb{I}\left[y_i = \hat{\mathcal{C}}(x_i)\right] \quad (8)$$

The **RCA** measure could be low because of two reasons. It could be due to the domain-shift we try to capture, as a very different distribution would have a major impact in the training data generated on top of D_t. Or it could be due to the accumulation of error created by the “back-and-forth” training. If D_t follows the same distribution than D_s, than the measure would only capture the impact of that accumulation of errors. To remove this source of errors, we also propose another measure **RCA⋆** which is the performance difference of $\hat{\mathcal{C}}$ and a classifier $\mathcal{C}'$ trained in the same way but using as target domain held-out data from the source domain. $\mathcal{C}$ is used again to pseudo-label a dataset, which is this time taken from the same distribution than D_s, and this new dataset is then used as training data for $\mathcal{C}'$. **RCA⋆** is then calculated as follows:

$$\textbf{RCA}\star = \frac{\mathit{1}}{|D_s^{'}|} \sum_{x_i, y_i \in D_s^{'}}^{m^{'}} \mathbb{I}\left[y_i = \mathcal{C}'(x_i)\right] - \mathbb{I}\left[y_i = \hat{\mathcal{C}}(x_i)\right] \quad (9)$$

A schematic view of these two measures is depicted in Fig. 2.



(a) **RCA**

(b) **RCA⋆**

Figure 2: The classifier $\mathcal{C}$ is trained on source domain (blue) with true labels (green) and is applied on target domain data (red) to create pseudo labels (grey). In the case of **RCA⋆** $\mathcal{C}$ is also applied on a held-out source domain data (blue). Pseudo-labels (grey) are then used to train new classifiers ($\hat{\mathcal{C}}$ and for **RCA⋆** also $\mathcal{C}'$). They are later applied on a test set of the source domain to calculate **RCA** and **RCA⋆**.

3 Experiments

3.1 Regression of Performance Drop

We present a regression based method that can directly estimate the performance drop of a model trained on D_s and tested on D_t. This method does not require any labeling in D_t however it assumes the availability of a small fixed number of labeled evaluation datasets $D_o \in D \setminus \{D_s, D_t\}$.

For each one of these fixed evaluation datasets, using simple linear regression we fit a regression line between the drop in the model accuracy and a domain-shift detection metrics of choice (§2). Afterwards, by calculating the value of this domain-shift detection metric on d_t we can then use this regression line to predict the performance drop when evaluating the model d_t.

In our experiments we report the Mean Absolute

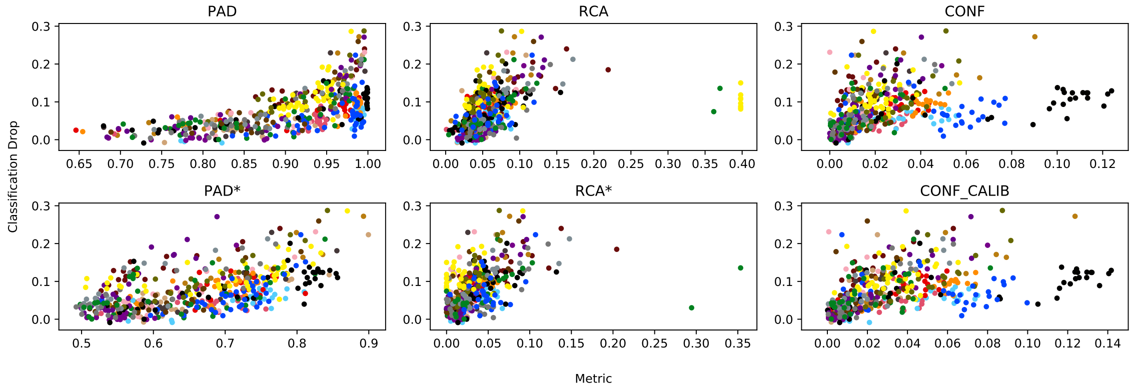


Figure 3: Scatter plots showing correlations between the values of each proposed domain similarity metric (x-axis) and the actual classification drop (y-axis) for the sentiment classification task. Each point corresponds to a single domain-shift scenario. Different colors represent different source domains and hence a single trained model on the source domain dataset.

Error (MAE) and Max error between our predicted values of performance drop using our method and the actual performance drop on d_t.

To put in perspective the predictive power of this regression method on each proposed metric, we implement a baseline (**`Mean`**) that instead of learning a regression it always gives the average classification drop over all evaluation datasets in D_o.

We also experiment with doing linear regression using all metrics proposed in §2 at once (**`Ensemble`**).

3.2 Datasets

We evaluate our metrics across two tasks of different nature namely sentiment analysis and part-of-speech (POS) tagging.

3.2.1 Sentiment Analysis

For sentiment analysis we follow (Blitzer et al., 2007; Ruder and Plank, 2018) by using the Amazon multidomain reviews dataset in English.[5] Although this dataset contains several domains they still come from the same platform which can restrain some diversity. To alleviate this we combine it with two other datasets namely Yelp[6] and IMDB movie reviews dataset[7]. Although the preprocessed dataset in (Blitzer et al., 2007) is widely used, it only consists of 4 domains and the input documents are reduced to their `TF-IDF` weights which is not a suitable input for modern NLP architectures. Thus we perform a different preprocessing across a wider range of domains as follows: After removing redundant reviews, we preprocess the dataset to obtain binary labels such that reviews with 1 to 3 stars are labeled as negative while reviews with 4 or 5 stars are labeled as positive. Finally we randomly sample 21K reviews (10K train, 10K valid and 1K test) from 21 domains. Yelp and IMDB datasets follow the same preprocessing steps and are added as 2 extra domains. This yield a total new dataset with 23 domains for sentiment analysis yielding 506 domain-shift scenarios.[8]

3.2.2 Part of Speech Tagging

For part of speech tagging we select 4 publicly available [9] Universal Dependencies datasets for English (Nivre et al., 2016). We split the EWT dataset (UD for English web treebank) according to each sub-category, while keeping the rest of the smaller datasets as is. This yields in total 8 domains with roughly comparable sizes ($\sim 4K$ sentences each) yielding in total 56 domain-shift scenarios.

3.3 Experiment Setup and Training Details

For each domain-shift scenario, the task model is trained on the source domain training split. Testing is performed on both source and target domains test sets. Simultaneously, we calculate each of our proposed metrics in §2. Note here that some of those metrics such as **`PAD`**, **`PAD*`**, **`RCA`**

[5] http://jmcauley.ucsd.edu/data/amazon/index.html

[6] https://www.yelp.com/dataset

[7] https://ai.stanford.edu/ amaas/data/sentiment/

[8] This makes 10x more examples and 40x more domain-shift scenarios than the widely used version of (Blitzer et al., 2007).

[9] ParTUT, GUM, EWT, LinES

and **RCA⋆** require the text of the target domain test set. None of the proposed metrics require any labels from the target domain, in line with the unsupervised scenario we are considering. The initial word embeddings are a hyperparameter, and we consider random initialization, pretrained GloVe (Pennington et al., 2014) with several dimensions and contextualized word embeddings using ELMo (Peters et al., 2018). As model architectures we use Multi-Layer Bi-LSTM (Graves and Schmidhuber, 2005) followed by a multi-layer feed-foward NN and a softmax. In sentiment analysis the feed forward network is applied on the last output of the Bi-LSTM to produce one label prediction for the whole sentence, while for POS tagging it is applied to each output to produce a label prediction for each corresponding token.

For training the domain classifiers used to calculate the **PAD** and **PAD⋆** metrics, we use similar model architecture as in sentiment analysis, initialized from scratch in case of **PAD** or initialized with the weights of the best task model in case of **PAD⋆**. Afterwards, they are trained to discriminate between inputs of the source and target domain datasets.

To calculate **CONF_CALIB**, the best performing model is selected and its confidence weights are calibrated using temperature scaling on the source domain validation set.

Each model is trained using Adam optimzation (Kingma and Ba, 2015) and early stopping with patience 5 over the source domain validation set.

All models and training code have been implemented using AllenNLP (Gardner et al., 2018) and made publicly available in addition to the Datasets.[10] The detailed hyper-parameters and the test results for each source domain dataset are shown in the appendix.

4 Evaluation Results

Fig. 3 contains a plot for each of the proposed metrics, showing the value of that measure vs the actual drop in performance for the sentiment analysis task. Each point corresponds to a single domain-shift scenario. While there is an overall trend between each of our proposed metrics and the actual drop in classification accuracy, the actual trend differs depending on which source domain was used.[11] This is unsurprising as all measures but **PAD** are model-specific. Instead of computing overall correlation trends, we decide to evaluate the capacity of each measure to serve as a predictor of the classification drop, as detailed in the next section.

4.1 Error in Performance Drop Prediction

Table 1 shows the mean absolute and maximum error values of the regression process that predicts the performance drop of a model trained over d_s and evaluated on d_t. The baseline that predicts always the mean performance drop achieves on a mean absolute error of 5.2% and max error of 12.77% for sentiment analysis, while this number drops for POS tagging to a mean of 1.06% and max of 1.67% in the worse case. All our proposed metrics improve significantly over that, with **PAD⋆** clearly improving over all other in both datasets. Overall, the best performing method is **PAD⋆** with 2.15% and 0.88% mean absolute error in prediction of performance drop for sentiment analysis and POS tagging respectively. Learning an ensemble between all metrics does not guarantee to provide the best predictions, which could be due to the small size of the points used for regression.

4.2 Impact of Number of Domains

Our proposed method for detecting performance drop assumes the existence of several evaluation datasets from different domains. This is a non-negligible cost, as having so many evaluation dataset from different domains might not be realistic in many scenarios. In this section we evaluate the impact of having a lower number of source domains from which to learn the classification drop. We sample randomly a smaller number of datasets, and repeat the experiment. In Fig. 4 and Fig. 5 we report the results of that with 5 different runs of sampling. As expected, the error decreases by increasing the number of out of domain test sets. However, prediction error enters an acceptable score with only 3 annotated source domains.

4.3 Adversarial Shift

PAD⋆ and **PAD** are calculated solely from learning to classify between the source and target domains and could therefore be particularly sensitive to task irrelevant domain-shifts i.e. input sig-

[10] https://github.com/hadyelsahar/domain-shift-prediction

[11] Each source domain corresponds to a single trained model, denoted by the same color in Fig. 3.

	Sentiment		POS tagging	
	MAE	Max	MAE	Max
Mean	5.2 ± 2.02	12.77	1.06 ± 0.36	1.67
RCA	$\underline{2.88} \pm \underline{1.31}$	$\underline{7.17}$	1.08 ± 0.31	1.58
RCA⋆	2.92 ± 1.39	7.42	$\underline{1.05} \pm \underline{0.27}$	$\underline{1.42}$
CONF	2.85 ± 1.69	8.86	$\underline{\mathbf{0.89}} \pm \underline{0.39}$	$\underline{1.75}$
CONF_CALIB	$\underline{2.67} \pm \underline{1.49}$	$\underline{8.13}$	1.12 ± 0.45	1.83
PAD	2.51 ± 1.54	8.16	1.24 ± 0.37	1.7
PAD⋆	$\underline{\mathbf{2.15}} \pm \underline{\mathbf{0.88}}$	$\underline{\mathbf{4.64}}$	$\underline{\mathbf{0.89}} \pm \underline{\mathbf{0.1}}$	$\underline{\mathbf{0.99}}$
Ensemble	2.22 ± 0.9	5.02	1.03 ± 0.42	1.78

Table 1: Mean absolute error (MAE) and max error (Max) of the performance drop prediction for the task of sentiment analysis (left) and POS tagging (right). Best results (the lower the better) are shown in **bold** while best results within each family of measure are underlined.

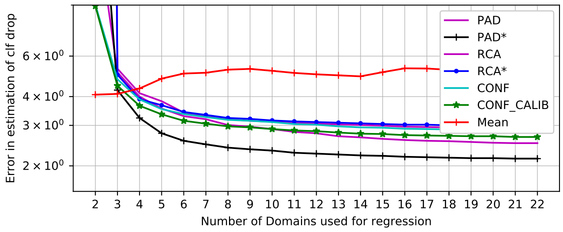


Figure 4: Fig. showing effect of number of domains on the prediction of accuracy drop for sentiment analysis.

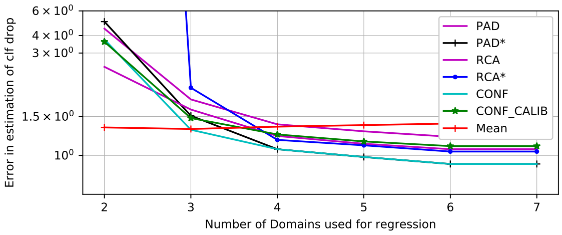


Figure 5: Fig. showing effect of number of domains on the prediction of accuracy drop for POS tagging.

	MAE	max
Mean	4.35 ± 2.31	10.79
RCA	$\underline{3.63} \pm 1.95$	9.56
RCA⋆	$3.67 \pm \underline{1.93}$	$\underline{8.44}$
CONF	2.88 ± 1.16	5.5
CONF_CALIB	$\underline{\mathbf{2.81}} \pm \underline{\mathbf{1.09}}$	$\underline{\mathbf{5.48}}$
PAD	$\underline{4.35} \pm 2.31$	10.79
PAD⋆	$4.6 \pm \underline{2.14}$	$\underline{10.77}$
Ensemble	3.02 ± 0.98	5.88

Table 2: Mean absolute error and Max error of performance drop prediction for sentiment analysis under adversarial shift evaluation (The lower the better).

nals that can help to differentiate domains apart have has no impact on the task itself. To evaluate the robustness of each of our proposed metrics in this specific scenario, we perform an experiment by applying an adversarial domain-shift. For each domain-shift scenario in the sentiment analysis task we add a different unique tag <SOURCE> and <TARGET> in the beginning of each example in the source and the target domains respectively: this has no impact on the final task classification, but makes it trivial to discriminate between the two domains. The results of re-running the same experiment on this modified dataset are in Table 2. As expected, **PAD** is greatly affected, not performing better than the baseline which just predicts the mean classification drop. The other two families are less or not at all affected. Surprisingly, **PAD⋆** also degrades significantly, despite using a task representation which should learn to discard the useless (for the task prediction) newly introduced token. To understand this better, we analyze the behaviour of the models with different depths. In Fig. 7 the best performing model for a given depth is used: the results indicate that the capacity for predicting the classification drop using **PAD⋆** is greatly influenced only if the model is very shallow. This becomes clearer in Fig. 6 where we repeat the plots from Fig. 3 for this adversarial dataset restricting the hyper-paramater search of the task model to models of a fixed depths. At any model depth the **PAD** measure is always maximal (1.00) as it learns to differentiate perfectly the two domains. While this is also the case for **PAD⋆** in the case of model of depth 2, deeper model are less sensitive to that. This might indicate that the higher layer (which are used as input representation for the domain classification models used to calculate **PAD⋆**) are learning to ignore the domain token as it is irrelevant for the task at hand. The rest of the metrics are more robust with respect to adversarial examples and model depth, which might make them good candidates for cases of severe co-variate shift.

5 Related Work

A large body of work has tackled the problem of defining, measuring and adapting to domain-shift in machine learning and NLP (Quionero-Candela et al., 2009; Blitzer and Pereira, 2007;

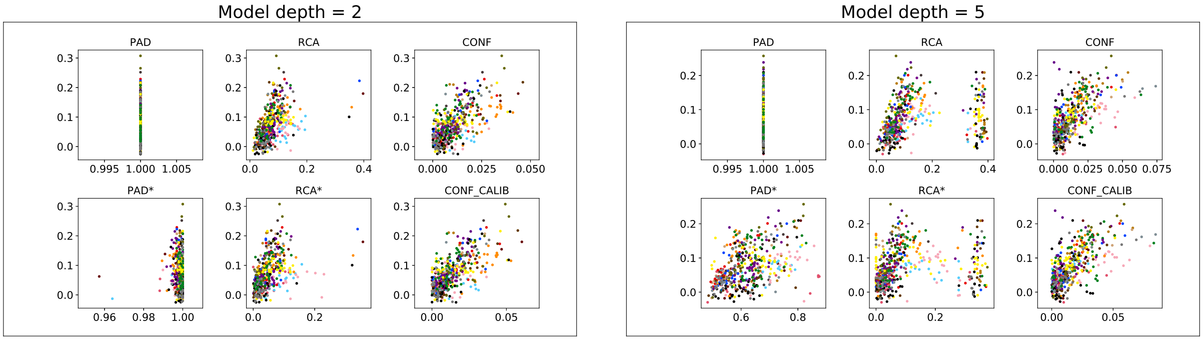


Figure 6: Measure vs drop in performance in the adversarial case, restricting the hyper-parameter search to models of depth 2 (left) and 5 (right)

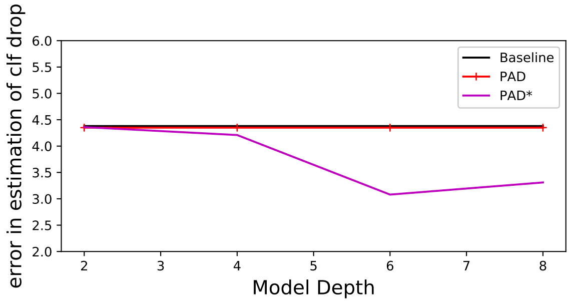


Figure 7: The effect of the task model depth on the capacity of the metric to estimate the drop in classification under adversarial evaluation.

Ben-David et al., 2010). Kifer et al. (2004); Ben-David et al. (2006) formalize the concept of $\mathcal{H}$-divergence to measure domain discrepancy. Later on, Blitzer et al. (2008) introduce the Proxy $\mathcal{A}$-distance as a proxy to $\mathcal{H}$-divergence which was used by Blitzer et al. (2007); Rai et al. (2010) to gain insights about adaptability of representations for domain adaptation and active learning. More recently, Ruder et al. (2017) showed that Proxy $\mathcal{A}$-distance is effective as a similarity metric for dataset selection. $\mathcal{H}$-divergence has inspired a large body of work (Ganin et al., 2016; Tzeng et al., 2017) for domain adaptation by minimizing the $\mathcal{H}$-divergence between domains using a domain adversarial loss.

There exists a large line of work on developing domain similarity metrics for data selection under transfer learning (Moore and Lewis, 2010; Plank and van Noord, 2011; Axelrod et al., 2011; Wu and Huang, 2016), however there is no consensus on which similarity measure is suitable for which NLP task. Ruder and Plank (2017) show that a combination of several of those similarity features re-weighted using Bayesian Optimization performs the best across several tasks. This method however is not fully unsupervised as it requires an annotated validation set from each target domain.

The closest work to ours are Ravi et al. (2008); Van Asch and Daelemans (2010), as they also tackle the problem of predicting performance of models at test time. They introduce a set of domain similarity metric that correlates with the performance of a model on a test set for both part-of-speech tagging and parsing. However, many of those measure are built on top of heavily hand-crafted features and evaluated through shallow models which are different to the way modern NLP models are built now.

Confidence scores can be a good estimation of domain similarity, however it is well known for modern neural networks that their confidence scores are usually mis-calibrated. Because of that, many self-training techniques rely on an ensemble of models to calculate for bootstrapping in-domain examples training examples, such as co-training and tri-training (Zhou and Li, 2005), tri-training with dis-agreement (Søgaard, 2010) and multi-task tri-training (Ruder and Plank, 2018). In our paper we aim to measure the potential classification drop of a specific model due to domain-shift. Training an ensemble of models of the same architecture in hand might be expensive and inefficient for just measuring domain-shift. We try to overcome this problem by calibrating confidence scores (Zadrozny and Elkan, 2002; Guo et al., 2017; DeVries and Taylor, 2018). Using confidence scores of calibrated models has shown a large success in out of distribution detection (Hendrycks and Gimpel, 2017; Liang et al., 2018, 2017; Lee et al., 2018).

The idea of reverse classification accuracy (Fan and Davidson, 2006; Zhong et al., 2010) was first introduced as a part of "Transfer Cross Validation" to select both models and data in a cross validation framework, optimized for transfer learning. More recently, this was adapted as a confidence estimator to predicting segmentation performance in the clinical domain (Valindria et al., 2017). The same idea has been used recently used for evaluating GANs for image generation (Shmelkov et al., 2018).

6 Conclusion

In this paper we studied the problem of prediction of performance drop due to domain-shift for modern NLP models, having no labeled target domain data but at least few fixed evaluation datasets from other domains. We investigated three family of metrics for measuring domain similarity. In each of them we introduced a novel adaptation on existing metrics to adapt to the different nature of modern NLP models and possibly obtain higher prediction scores of the performance drop. Our evaluation over two NLP tasks show that this drop can be estimated very accurately even when only few other source-domains evaluation datasets are available. In general, the family of $\mathcal{H}$-divergence based measures perform the best. However, they are prone to fail when there is a severe change in the marginal distribution that is task irrelevant. In particular, the well established **PAD** measure is rendered useless in a setting where we artificially exaggerate that phenomena. Using a task-specific representation is slightly more robust to that problem, although only if a deeper model is used.

A strong family of measures that does not have that drawback are confidence-based measures, but they require access to the confidence weights and not only the predicted labels by the model. Unfortunately, our adaptation of the reverse classification accuracy measure **RCA*** does not obtain higher performance than the simple **RCA**. This could be due to the fact that datasets sampled from the same domain can still be subjected to sample selection bias. In the future work, we plan to investigate ways to solve that. These conclusions are summarized in Table 3, which we recommend as guideline when deciding what measure to use.

Measure	Robust to co-variate shift	Black box	Good with small Target Data
PAD	X	✓	X
PAD*	(X)	X	X
RCA	✓	✓	X
RCA*	✓	✓	X
CONF	✓	X	✓
CONF_CALIB	✓	X	✓

Table 3: Summary of domain similarity metrics discussed in this work and their different characteristics.

Acknowledgments

This work was partially funded through the European Unions Horizon 2020 research and innovation programme under grant agreement No. 786741 (SMOOTH Platform).

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. ACM.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 137–144.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. 2008. Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pages 129–136.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

John Blitzer and Fernando Pereira. 2007. Domain adaptation of natural language processing systems. *University of Pennsylvania*, pages 1–106.

Caroline Brun and Vassilina Nikoulina. 2018. Aspect based sentiment analysis into the wild. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2018, Brussels, Belgium, October 31, 2018*, pages 116–122.

Terrance DeVries and Graham W. Taylor. 2018. Learning confidence for out-of-distribution detection in neural networks. *CoRR*, abs/1802.04865.

Wei Fan and Ian Davidson. 2006. Reverse testing: an efficient framework to select amongst classifiers under sample selection bias. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 147–156. ACM.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:59:1–59:35.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1321–1330.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Daniel Kifer, Shai Ben-David, and Johannes Gehrke. 2004. Detecting change in data streams. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, pages 180–191.

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

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Shiyu Liang, Yixuan Li, and R. Srikant. 2017. Principled detection of out-of-distribution examples in neural networks. *CoRR*, abs/1706.02690.

Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Zachary C. Lipton, Yu-Xiang Wang, and Alexander J. Smola. 2018. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3128–3136. PMLR.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA*.

Robert C. Moore and William D. Lewis. 2010. Intelligent selection of language model training data. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden, Short Papers*, pages 220–224. The Association for Computer Linguistics.

Jose G Moreno-Torres, Troy Raeder, RocíO Alaiz-RodríGuez, Nitesh V Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan T. McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Barbara Plank and Gertjan van Noord. 2011. Effective measures of domain similarity for parsing. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1566–1576. The Association for Computer Linguistics.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2009. *Dataset shift in machine learning*. The MIT Press.

Piyush Rai, Avishek Saha, Hal Daumé III, and Suresh Venkatasubramanian. 2010. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 27–32. Association for Computational Linguistics.

Sujith Ravi, Kevin Knight, and Radu Soricut. 2008. Automatic prediction of parser accuracy. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 887–896. ACL.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2017. Data selection strategies for multi-domain sentiment analysis. *CoRR*, abs/1702.02426.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 372–382. Association for Computational Linguistics.

Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1044–1054.

Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. 2018. How good is my gan? In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, volume 11206 of *Lecture Notes in Computer Science*, pages 218–234. Springer.

Anders Søgaard. 2010. Simple semi-supervised training of part-of-speech taggers. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden, Short Papers*, pages 205–208.

Amos Storkey. 2009. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, pages 3–28.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2962–2971. IEEE Computer Society.

Vanya V Valindria, Ioannis Lavdas, Wenjia Bai, Konstantinos Kamnitsas, Eric O Aboagye, Andrea G Rockall, Daniel Rueckert, and Ben Glocker. 2017. Reverse classification accuracy: predicting segmentation performance in the absence of ground truth. *IEEE transactions on medical imaging*, 36(8):1597–1606.

Vincent Van Asch and Walter Daelemans. 2010. Using domain similarity for performance estimation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 31–36. Association for Computational Linguistics.

Fangzhao Wu and Yongfeng Huang. 2016. Sentiment domain adaptation with multiple sources. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 694–699.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. 2013. Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 819–827. JMLR.org.

Erheng Zhong, Wei Fan, Qiang Yang, Olivier Verscheure, and Jiangtao Ren. 2010. Cross validation framework to choose amongst models and datasets for transfer learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 547–562. Springer.

Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. Knowl. Data Eng.*, 17(11):1529–1541.