PERL: Pivot-based Domain Adaptation for Pre-trained Deep Contextualized Embedding Models

Eyal Ben-David* **Carmel Rabinovitz*** **Roi Reichart**

Technion, Israel Institute of Technology

{eyalbd12@campus.|carmelrab@campus.|roiri@}technion.ac.il

Abstract

Pivot-based neural representation models have led to significant progress in domain adaptation for NLP. However, previous research following this approach utilize only labeled data from the source domain and unlabeled data from the source and target domains, but neglect to incorporate massive unlabeled corpora that are not necessarily drawn from these domains. To alleviate this, we propose *PERL*: A representation learning model that extends contextualized word embedding models such as BERT (Devlin et al., 2019) with pivot-based fine-tuning. PERL outperforms strong baselines across 22 sentiment classification domain adaptation setups, improves in-domain model performance, yields effective reduced-size models, and increases model stability.[1]

1 Introduction

Natural Language Processing (NLP) algorithms are constantly improving, gradually approaching human-level performance (Dozat and Manning, 2017; Edunov et al., 2018; Radford et al., 2018). However, those algorithms often depend on the availability of large amounts of manually annotated data from the domain in which the task is performed. Unfortunately, collecting such annotated data is often costly and laborious, which substantially limits the applicability of NLP technology.

Domain Adaptation (DA), training an algorithm on annotated data from a source domain so that it can be effectively applied to other target domains, is one of the ways to solve the above bottleneck. Indeed, over the years substantial efforts have been devoted to the DA challenge (Roark and Bacchiani, 2003; Daumé III and Marcu, 2006; Ben-David et al., 2010; Jiang and Zhai, 2007; McClosky et al., 2010; Rush et al., 2012; Schnabel and Schütze, 2014). Our focus in this paper is on unsupervised DA, the setup we consider most realistic. In this setup labeled data is available only from the source domain and unlabeled data is available from both the source and the target domains.

While various approaches for DA have been proposed (§2), with the prominence of deep neural network (DNN) modeling, attention has been recently focused on representation learning approaches. Within representation learning for unsupervised DA, two approaches have been shown particularly useful. In one line of work, DNN-based methods that use compress-based noise reduction to learn cross-domain features have been developed (Glorot et al., 2011; Chen et al., 2012). In another line of work, methods based on the distinction between pivot and non-pivot features (Blitzer et al., 2006, 2007) learn a joint feature representation for the source and the target domains. Later on, Ziser and Reichart (2017, 2018), and Li et al. (2018) married the two approaches and achieved substantial improvements on a variety of DA setups.

Despite their success, pivot-based DNN models still only utilize labeled data from the source domain and unlabeled data from both the source and the target domains, but neglect to incorporate massive unlabeled corpora that are not necessarily drawn from these domains. With the recent game-changing success of contextualized word embedding models trained on such massive corpora (Devlin et al., 2019; Peters et al., 2018), it is natural to ask whether information from such corpora can enhance these DA methods, particularly that background knowledge from non-contextualized embeddings has shown useful for

*Both authors contributed equally to this work.

[1]Our code is at https://github.com/eyalbd2/PERL.

Transactions of the Association for Computational Linguistics, vol. 8, pp. 504–521, 2020. https://doi.org/10.1162/tacl_a_00328
Action Editor: Jimmy Lin. Submission batch: 4/2020; Revision batch: 2020; Published 8/2020.

DA (Plank and Moschitti, 2013; Nguyen et al., 2015).

In this paper we hence propose an unsupervised DA approach that extends leading approaches based on DNNs and pivot-based ideas, so that they can incorporate information encoded in massive corpora (§3). Our model, named *PERL*: *Pivot-based Encoder Representation of Language*, builds on massively pre-trained contextualized word embedding models such as BERT (Devlin et al., 2019). To adjust the representations learned by these models so that they close the gap between the source and target domains, we fine-tune their parameters using a pivot-based variant of the Masked Language Modeling (MLM) objective, optimized on unlabeled data from both the source and the target domains. We further present R-PERL (regularized PERL), which facilitates parameter sharing for pivots with similar meaning.

We perform extensive experimentation in various unsupervised DA setups of the task of binary sentiment classification (§4, 5). First, for compatibility with previous work, we experiment with the legacy product review domains of Blitzer et al. (2007) (12 setups). We then experiment with more challenging setups, adapting between the above domains and the airline review domain (Nguyen, 2015) used in Ziser and Reichart (2018) (4 setups), as well as the IMDb movie review domain (Maas et al., 2011) (6 setups). We compare PERL to the best performing pivot-based methods (Ziser and Reichart, 2018; Li et al., 2018) and to DA approaches that fine-tune a massively pre-trained BERT model by optimizing its standard MLM objective using target-domain unlabeled data (Lee et al., 2020; Han and Eisenstein, 2019). PERL and R-PERL substantially outperform these baselines, emphasizing the additive effect of massive pre-training and pivot-based fine-tuning.

As an additional contribution, we show that pivot-based learning is effective beyond improving domain adaptation accuracy. Particularly, we show that an in-domain variant of PERL substantially improves the in-domain performance of a BERT-based sentiment classifier, for varying training set sizes (from 100 to 20K labeled examples). We also show that PERL facilitates the generation of effective reduced-size DA models. Finally, we perform an extensive ablation study (§6) that uncovers PERL's crucial design choices and demonstrates the stability of PERL to hyper-parameter selection compared to other DA methods.

2 Background and Previous Work

There are several approaches to DA, including instance re-weighting (Sugiyama et al., 2007; Huang et al., 2006; Mansour et al., 2008), subsampling from the participating domains Chen et al. (2011) and DA through representation learning, where a joint representation is learned based on texts from the source and target domains (Blitzer et al., 2007; Xue et al., 2008; Ziser and Reichart, 2017, 2018). We first describe the unsupervised DA pipeline, continue with representation learning methods for DA with a focus on pivot-based methods, and, finally, describe contextualized embedding models.

Unsupervised Domain Adaptation through Representation Learning As noted in §1 our focus in this work is on unsupervised DA through representation learning. A common pipeline for this setup consists of two steps: (A) Learning a representation model (often referred to as the encoder) using the source and target unlabeled data; and (B) Training a supervised classifier on the source domain labeled data. To facilitate domain adaptation, every text fed to the classifier in the second step is first represented by the pre-trained encoder. This is performed both when the classifier is trained in the source domain and when it is applied to new text from the target domain.

Exceptions to this pipeline are end-to-end models that jointly learn to perform the cross-domain text representation and the classification task. This is achieved by training a unified objective on the source domain labeled data and the unlabeled data from both the source and the target. Among these models are domain adversarial networks (Ganin et al., 2016), which were strongly outperformed by Ziser and Reichart (2018) to which we compare our methods, and the hierarchical attention transfer network (HATN; Li et al., 2018), which is one of our baselines (see below).

Unsupervised DA through representation learning has followed two main avenues. The first avenue consists of works that aim to explicitly build a feature representation that bridges the gap between the domains. A seminal framework in this line is structural correspondence learning (SCL; Blitzer et al., 2006, 2007), that splits the feature space into pivot and non-pivot features. A large

number of works have followed this idea (e.g., Pan et al., 2010; Gouws et al., 2012; Bollegala et al., 2015; Yu and Jiang, 2016; Li et al., 2017, 2018; Tu and Wang, 2019; Ziser and Reichart, 2017, 2018) and we discuss it below.

Works in the second avenue learn cross-domain representations by training autoencoders (AEs) on the unlabeled data from the source and target domains. This way they hope to obtain a more robust representation, which is hopefully better suited for DA. Examples for such models include the stacked denoising AE (SDA; Vincent et al., 2008; Glorot et al., 2011, the marginalized SDA and its variants (MSDA; Chen et al., 2012; Yang and Eisenstein, 2014; Clinchant et al., 2016) and variational AE based models (Louizos et al., 2016).

Recently, Ziser and Reichart (2017, 2018) and Li et al. (2018) married these approaches and presented pivot-based approaches where the representation model is based on DNN encoders (AE, long short-term memory [LSTM], or hierarchical attention networks). Because their methods outperformed the above models, we aim to extend them to models that can also exploit massive out of (source and target) domain corpora. We next elaborate on pivot-based approaches.

Pivot-based Domain Adaptation Proposed by Blitzer et al. (2006, 2007) through their SCL framework, the main idea of pivot-based DA is to divide the shared feature space of the source and the target domains to two complementary subsets: one of pivots and one of non-pivots. Pivot features are defined based on two criteria: (a) They are frequent in the unlabeled data of both domains; and (b) They are prominent for the classification task defined by the source domain labeled data. Non-pivot features are those features that do not meet at least one of the above criteria. While SCL is based on linear models, there have been some very successful recent efforts to extend this framework so that non-linear encoders (DNNs) are utilized. Here we focus on the latter line of work, which produces much better results, and do not elaborate on SCL any further.

Ziser and Reichart (2018) have presented the Pivot Based Language Model (PBLM), which incorporates pre-training and pivot-based learning. PBLM is a variant of an LSTM-based language model, but instead of predicting at each point the most likely next input word, it predicts the next input unigram or bigram if one of these is a pivot (if both are, it predicts the bigram), and NONE otherwise. In the unsupervised DA pipeline PBLM is trained on the source and target unlabeled data. Then, when the task classifier is trained and applied to the target domain, PBLM is used as a contextualized word embedding layer. Notice that PBLM is not pre-trained on massive out of (source and target) domain corpora, and its single-layer, unidirectional LSTM architecture is probably not ideal for knowledge encoding from such corpora.

Another work in this line is HATN (Li et al., 2018). This model automatically learns the pivot/non-pivot distinction, rather than following the SCL definition as Ziser and Reichart (2017, 2018) does. HATN consists of two hierarchical attention networks, P-net and NP-net. First, it trains the P-net on the source labeled data. Then, it decodes the most prominent tokens of P-net (i.e., tokens that received the highest attention values), and considers them as its pivots. Finally, it simultaneously trains the P-net and the NP-net on both the labeled and the unlabeled data, such that P-net is adversarially trained to predict the domain of the input example (Ganin et al., 2016) and NP-net is trained to predict its pivots, and the hidden representations from both networks serve for the task label (sentiment) prediction.

Both HATN and PBLM strongly outperform a large variety of previous DA models on various cross-domain sentiment classification setups. Hence, they are our major baselines in this work. Like PBLM, we use the same definition of the pivot and non-pivot subsets as in Blitzer et al. (2007). Like HATN, we also use an attention-based DNN. Unlike both models, we design our model so that it incorporates pivot-based learning with pre-training on massive out of (source and target) domain corpora. We next discuss this pre-training process, which is also known as training models for contextualized word embeddings.

Contextualized Word Embedding Models Contextualized word embedding (CWE) models are trained on massive corpora (Peters et al., 2018; Radford et al., 2019). They typically utilize a language modeling objective or a closely related variant (Peters et al., 2018; Ziser and Reichart, 2018; Devlin et al., 2019; Yang et al., 2019), although in some recent papers the model is trained on a mixture of basic NLP tasks (Zhang et al., 2019; Rotman and Reichart, 2019). The contribution

of such models to the state-of-the-art in a variety of NLP tasks is already well-established.

CWE models typically follow three steps: (1) Pre-training: Where a DNN (referred to as the encoder of the model) is first trained on massive unlabeled corpora which represent a broad domain (such as English Wikipedia); (2) Fine-tuning: An optional step, where the encoder is refined on unlabeled text of interest. As noted above, Lee et al. (2020) and Han and Eisenstein (2019) tuned BERT on unlabeled target domain data to facilitate domain adaptation; and (3) Supervised task training: Where task specific layers are trained on labeled data for a downstream task of interest.

PERL uses a pre-trained encoder, BERT in this paper. BERT's architecture is based on multi-head attention layers, trained with a two-component objective: (a) MLM and (b) Is-next-sentence prediction (NSP). For Step 2, PERL modifies only the MLM objective and it can hence be implemented within any CWE framework that uses this objective (Liu et al., 2019; Lan et al., 2020; Yang et al., 2019).

MLM is a modified language modeling objective, adjusted to self-attention models. When building the pre-training task, all input tokens have the same probability to be masked.[2] After the masking process, the model has to predict a distribution over the vocabulary for each masked token given the non-masked tokens. The input text may have more than one masked token, and when predicting one masked token information from the other masked tokens is not utilized.

In the next section we describe our PERL domain adaptation model. The novel component of this model is a pivot-based MLM objective, optimized at the fine-tuning step (Step 2) of the CWE pipeline, using source and target unlabeled data.

3 Domain adaptation with PERL

PERL uses pivot features in order to learn a representation that bridges the gap between two domains. Contrary to previous pivot-based DA representation models, it exploits unlabeled data from the source and target domains, and also from massive out of source and target domain corpora.

[2]We use the *huggingface* BERT code (Wolf et al., 2019): `https://github.com/huggingface/transformers`, where the masking probability is 0.15.

PERL consists of three steps that correspond to the three steps of CWE models, as described in § 2: *(1) Pre-training (Figure 1a)*: in which it utilizes a pre-trained CWE model (encoder, BERT in this work) that was trained on massive corpora; *(2) Fine-tuning (Figure 1b)*: where it refines some of the pre-trained encoder weights, based on a pivot-based objective that is optimized on unlabeled data from the source and target domains; and *(3) Supervised task training (Figure 1c)*: where task specific layers are trained on source domain labeled data for the downstream task of interest.

Our pivot selection method is identical to that of Blitzer et al. (2007) and Ziser and Reichart (2017, 2018). That is, the pivots are selected independently of the above three steps protocol.

We further present a variant of PERL, denoted with R-PERL, where the non-contextualized embedding matrix of the BERT model trained at Step (1) is used in order to regularize PERL during its fine-tuning stage (Step 2). We elaborate on this model towards the end of this section. We next provide a detailed description.

Pivot Selection Being a pivot-based language representation model, PERL is based on high quality pivot extraction. Since the representation learning is based on a masked language modeling task, the feature set we address consists of the unigrams and bigrams of the vocabulary. We base the division of this feature set into pivots and non-pivots on unlabeled data from the source and target domains. Pivot features are: (a) Frequent in the unlabeled data from the source and target domains; and (b) Among those frequent features, pivot features are the ones whose mutual information with the task label according to source domain labeled data crosses a pre-defined threshold. Features that do not meet the above two criteria form the non-pivot feature subset.

PERL pre-training (Step 1, Figure 1a) In order to inject prior language knowledge to our model, we first initialize the PERL encoder with a powerful pre-trained CWE model. As noted above, our rationale is that the general language knowledge encoded in these models, which is not specific to the source or target domains, should be useful for DA just as it has shown useful for in-domain learning. In this work we use BERT, although any other CWE model that employs

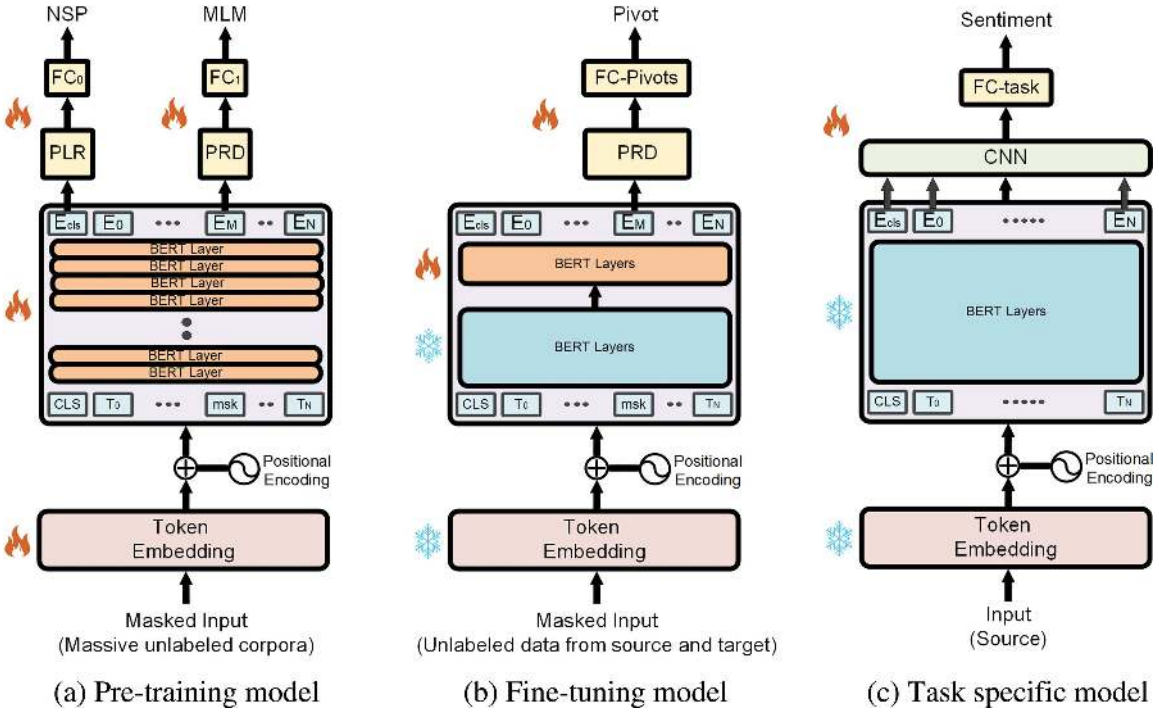

Figure 1: Illustrations of the three PERL steps. PRD and PLR stand for the BERT prediction head and pooler head, respectively, FC is a fully connected layer, and msk stands for masked tokens embeddings (embeddings of tokens that were masked). NSP and MLM are the next sentence prediction and masked language model objectives. For the definitions of the PRD and PRL layers as well as the NSP objective, see Devlin et al. (2019). We mark frozen layers (layers whose parameters are kept fixed) and non-frozen layers with snow-flake and fire symbols, respectively. The token embedding and BERT layers values at the end of each step initialize the corresponding layers of the next step model. The BERT box of the fine tuning step is described in more details in Figure 2.

the MLM objective for pre-training (Step 1) and fine-tuning (Step 2), could have been used.

PERL fine-tuning (Step 2, Figure 1b) This step is the core novelty of PERL. Our goal is to refine the initialized encoder on unlabeled data from the source and the target domains, using the distinction between pivot and non-pivot features.

For this aim we fine-tune the parameters of the pre-trained BERT using its MLM objective, but we choose the masked words so that the model learns to map non-pivot to pivot features. Recall that when building the MLM training task, each training example consists of an input text in which some of the words are masked, and the task of the model is to predict the identity of each of the masked words given the rest of the (non-masked) input text. Whereas in standard MLM training all input tokens have the same probability to be masked, in the PERL fine-tuning step we change both the masking probability and the prediction task so that the desired non-pivot to pivot mapping is learned. We next describe these two changes; see also a detailed graphical illustration in Figure 2.

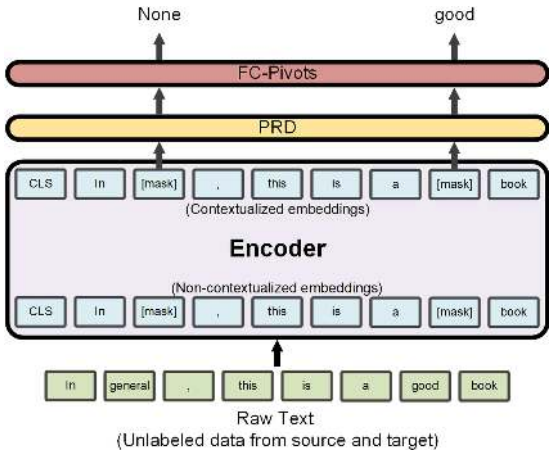

Figure 2: The PERL pivot-based fine-tuning task (Step 2). In this example two tokens are masked, *general* and *good*, only the latter is a pivot. The architecture is identical to that of BERT but the MLM task and the masking process are different, taking into account the pivot/non-pivot distinction.

1. *Prediction task.* While in standard MLM the task is to predict a token out of the entire vocabulary, here we define a pivot-base prediction task. Particularly, the model

should predict whether the masked token is a pivot feature or not, and if it is then it has to identify the pivot. That is, this is a multi-class classification task where the number of classes is equal to the number of pivots plus 1 (for the non-pivot prediction).

Put more formally, the modified pivot-based MLM objective is:

$$p(y_i = j) = \frac{e^{f(h_i)\cdot W_j}}{\sum_{k=1}^{|P|} e^{f(h_i)\cdot W_k} + e^{f(h_i)\cdot W_{none}}}$$

where y_i is a masked unigram or bigram at position i, P is the set of pivot features (token unigrams and bigrams), h_i is the encoder representation for the i-th token, W (the FC-Pivots layer of Figure 1b and Figure 2) is the pivot predictor matrix that maps from the latent space to the pivot set space (W_a is the a-th row of W), and f is a non-linear function composed of a *dense* layer, a *gelu* activation layer and *LayerNorm* (the PRD layer of Figure 1b and Figure 2).

2. *Masking process.* Instead of masking each input token (unigram) with the same probability, we perform the following masking process. For each input token (unigram) we first check whether it forms a bigram pivot together with the next token, and if so we mask this bigram with a probability of α. If the answer is negative, we check if the token at hand is a unigram pivot and if so we again mask it with a probability of α. Finally, if the token is not a pivot we mask it with a probability of β. Our hyper-parameter tuning process revealed that the values of $\alpha = 0.5$ and $\beta = 0.1$ provide strong results across our various experimental setups (see more on this in §6). This way PERL gives a higher probability to pivot masking, and by doing so the encoder parameters are fine-tuned so that they can predict (mostly) pivot features based (mostly) on non-pivot input.

Designing the fine-tuning task this way yields two advantages. First, the model should shape its parameters so that most of the information about the input pivots is preserved, while most of the information preserved about the non-pivots is what needed in order to predict the existence of the pivots. This way the model keeps mostly the information about unigrams and bigrams that are shared among the two domains and are significant for the supervised task, thus hopefully increasing its cross-domain generalization capacity.

Second, standard MLM, which has recently been used for fine-tuning in domain adaptation (Lee et al., 2020; Han and Eisenstein, 2019), performs a multi-class classification task with 30K tokens,[3] which requires $\sim$ 23M parameters as in the FC1 layer of Figure 1. By focusing PERL on pivot prediction, we can use only a factor of $\frac{|P|+1}{30K}$ of the FC layer parameters, as we do in the FC-pivots layer (Figure 1, where $|P|$ is the number of pivots, in our experiments $|P| \in [100, 500]$).

Supervised task training (Step 3, Figure 1c) To adjust PERL for a downstream task, we place a classification network on top of its encoder. While training on labeled data from the source domain and testing on the target domain, each input text is first represented by the encoder and is then fed to the classification network. Because our focus in this work is on the representation learning, the classification network is kept simple, consisting of one convolution layer followed by an average pooling layer and a linear layer. When training for the downstream task, the encoder weights are frozen.

R-PERL A potential limitation of PERL is that it ignores the semantics of its pivots. While the negative pivots *sad* and *unhappy* encode similar information with respect to the sentiment classification task, PERL considers them as two different output classes. To alleviate this, we propose the regularized PERL (R-PERL) model where pivot-similarity information is taken into account.

To achieve this we construct the FC-pivots matrix of R-PERL (Figure 1b and 2) based on the Token Embedding matrix learned by BERT in its pre-training stage (Figure 1a). Particularly, we fix the unigram pivot rows of the FC-pivots matrix to the corresponding rows in BERT's Token Embedding matrix, and the bigram pivot rows to the mean of the Token Embedding rows that correspond to the unigrams that form this bigram. The FC-pivots matrix of R-PERL is kept fixed during fine-tuning.

Our assumptions are that: (1) Pivots with similar meaning, such as *sad* and *unhappy*, have similar representations in the Token Embedding matrix

[3]The BERT implementation we use keeps a fixed 30K word vocabulary, derived from its pre-training process.

learned at the pre-training stage (Step 1); and (2) There is a positive correlation between the appearance of such pivots (i.e., they tend to appear, or not appear, together; see Ziser and Reichart [2017] for similar considerations). In its fine-tuning step, R-PERL is hence biased to learn similar representations to such pivots in order to capture the positive correlation between them. This follows from the fact that pivot probability is computed by taking the dot product of its representation with its corresponding row in the FC-pivots matrix.

4 Experiments

Tasks and Domains Following a large body of prior DA work, we focus on the task of binary sentiment classification. For compatibility with previous literature, we first experiment with the four legacy product review domains of Blitzer et al. (2007): Books (B), DVDs (D), Electronic items (E), and Kitchen appliances (K), with a total of 12 cross-domain setups. Each domain has 2,000 labeled reviews, 1,000 positive and 1,000 negative, and unlabeled reviews as follows: B: 6,000, D: 34,741, E: 13,153 and K: 16,785.

We next experiment in a more challenging setup, considering an airline review dataset (A) (Nguyen, 2015; Ziser and Reichart, 2018). This setup is challenging both due to the differences between the product and service domains, and because the prior probability of observing a positive review at the A domain is much lower than the same probability in the product domains.[4] For the A domain, following Ziser and Reichart (2018), we randomly sampled 1,000 positive and 1,000 negative reviews for our labeled set, and 39,396 reviews for our unlabeled set. Due to the heavy computational demands of the experiments, we arbitrarily chose 3 product to airline and 3 airline to product setups.

We further consider an additional modern domain: IMDb (I) (Maas et al., 2011),[5] which is commonly used in recent sentiment analysis work. This dataset consists of 50,000 movie reviews from IMDb (25,000 positive and 25,000 negative), where there is a limitation on the number of reviews per movie. We randomly sampled 2,000 labeled reviews, 1,000 positive and 1,000 negative, for our labeled set, and the remaining 48,000 reviews form our unlabeled set.[6] As above, we arbitrarily chose 2 IMDb to product and 2 product to IMDb setups for our experiments.

Pivot-based representation learning has shown instrumental for DA. We hypothesize that it can also be beneficial for in-domain tasks, as it focuses the representation on the information encoded in prominent unigrams and bigrams. To test this hypothesis we experiment in an in-domain setup, with the IMDb movie review dataset. We follow the same experimental setup as in the domain adaptation case, except that only IMDb unlabeled data is used for fine-tuning, and the frequency criterion in pivot selection is defined with respect to this dataset.

We randomly sampled 25,000 training and 25,000 test examples, keeping the two sets balanced, and additional 50,000 reviews formed an unlabeled balanced set.[7] We consider 6 setups, differing in their training set size: 100, 500, 1K, 2K, 10K, and 20K randomly sampled examples.

Baselines We compare our PERL and R-PERL models to the following baselines: (a+b) PBLM-CNN and PBLM-LSTM (Ziser and Reichart, 2018), differing only in their classification layer (CNN vs. LSTM);[8] (c) HATN (Li et al., 2018);[9] (d) BERT; and (e) Fine-tuned BERT (following Lee et al., 2020 and Han and Eisenstein, 2019): This model is identical to PERL, except that the fine-tuning stage is performed with a standard MLM instead of our pivot-based MLM. BERT, Fine-tuned BERT, PBLM-CNN, PERL, and R-PERL all use the same CNN-based sentiment classifier, while HATN jointly learns the feature representation and performs sentiment classification.

Cross-validation We use a five-fold cross-validation protocol, where in every fold 80% of the source domain examples are randomly selected for training data, and 20% for development data (both sets are kept balanced). For each model we report the average results across the five folds. In each fold we tune the hyper-parameters so that

[4]This analysis, performed by Ziser and Reichart (2018), is based on the gold labels of the unlabeled data.

[5]The details of the IMDb dataset are available at: http://www.andrew-maas.net/data/sentiment.

[6]We make sure that all reviews of the same movie appear either in the training set or in the test set.

[7]These reviews are also part of the IMDb dataset.

[8]https://github.com/yftah89/PBLM-Domain-Adaptation.

[9]https://github.com/hsqmlzno1/HATN.

to minimize the cross-entropy development data loss.

Hyper-parameter Tuning For all models we use the WordPiece word embeddings (Wu et al., 2016) with a vocabulary size of 30k, and the same optimizer (with the same hyper-parameters) as in their original paper. For all pivot-based methods we consider the unigrams and bigrams that appear at least 20 times both in the unlabeled data of the source domain and in the unlabeled data of the target domain as candidates for pivots,[10] and from these we select the $|P|$ candidates with the highest mutual information with the task source domain label ($|P| = \{100, 200, \ldots, 500\}$). The exception is HATN that automatically selects its pivots, which are limited to unigrams.

We next describe the hyper-parameters of each of the models. Due to our extensive experimentation (22 DA and 6 in-domain setups, 5-fold cross-validation), we limit our search space, especially for the heavier components of the models.

R-PERL, PERL, BERT and Fine-tuned BERT For the encoder, we use the BERT-base uncased architecture with the same hyper-parameters as in Devlin et al. (2019), tuning for PERL, R-PERL and Fine-tuned BERT the number of fine-tuning epochs (out of: 20, 40, 60) and the number of unfrozen BERT layer during the fine-tuning process (1, 2, 3, 5, 8, 12). For PERL and R-PERL we tune the number of pivots (100, 200, 300, 400, 500) as well as α and β (0.1, 0.3, 0.5, 0.8). The supervised task classifier is a basic CNN architecture, which enables us to search over the number of filters (out of: 16, 32, 64), the filter size (7, 9, 11) and the training batch size (32, 64).

PBLM-LSTM and PBLM-CNN For PBLM we tune the input word embedding size (32, 64, 128, 256), the number of pivots (100, 200, 300, 400, 500), and the hidden dimension (128, 256, 512). For the LSTM classification layer of PBLM-LSTM we consider the same hidden dimension and input word embedding size as for the PBLM encoder. For the CNN classification layer of PBLM-CNN, following Ziser and Reichart (2018) we use 250 filters and a kernel size of 3. In each setup we choose the PBLM model (PBLM-LSTM or PBLM-CNN) that yields better test set accuracy and report its result, under PBLM-Max.

[10]In the in-domain experiments we consider the IMDb unlabeled data.

HATN The hyper-parameters of Li et al. (2018) were tuned on a larger training set than ours, and they hence yield sub-optimal performance in our setup. We tune the training batch size (20, 50 300), the hidden layer size (20, 100, 300), and the word embedding size (50, 100, 300).

5 Results

Overall results Table 1 presents domain adaptation results, and is divided to two panels. The top panel reports results on the 12 setups derived from the 4 legacy product review domains of Blitzer et al. (2007) (denoted with $P \Leftrightarrow P$). The bottom panel reports results for 10 setups involving product review domains and the IMDb movie review domain (left side; denoted $P \Leftrightarrow I$) or the airline review domain (right side; denoted $P \Leftrightarrow A$). Table 2 presents in-domain results on the IMDb domain, for various training set sizes.

Domain Adaptation As presented in Table 1, PERL models are superior in 20 out of 22 DA setups, with R-PERL performing best in 17 out of 22 setups. In the $P \Leftrightarrow P$ setups, their averaged performance (top table, All column) are 87.5% and 86.9% (for R-PERL and PERL, respectively) compared with 82.3% of HATN and 80.7% of PBLM-Max. Importantly, in the more challenging setups, the performance of one of these baselines substantially degrade. Particularly, the averaged R-PERL and PERL performance in the $P \Leftrightarrow I$ setups are 84.7% and 84.4%, respectively (bottom panel, left All column), compared with 75.5% of HATN and 69.0% of PBLM-Max. In the $P \Leftrightarrow A$ setups the averaged R-PERL and PERL performances are 84.2% and 82.9%, respectively (bottom panel, right All column), compared with 80.5% of PBLM-Max and only 71.8% of HATN.

The performance of BERT and Fine-tuned BERT also degrade on the challenging setups: From an average of 80.2% (BERT) and 84.1% (Fine-tuned BERT) in $P \Leftrightarrow P$ setups, to 74.2% and 78.9%, respectively, in $P \Leftrightarrow I$ setups, and to 75.6% and 79.4%, respectively, in $P \Leftrightarrow A$ setups. R-PERL and PERL, in contrast, remain stable across setups, with an averaged accuracy of 84.2–87.5% (R-PERL) and 82.9–86.8% (PERL).

The IMDb and airline domains differ from the product domains in their topic (movies [IMDb] and services [airline] vs. products). Moreover, the unlabeled data from the airline domain contains

	D → K	D → B	E → D	B → D	B → E	B → K	E → B	E → K	D → E	K → D	K → E	K → B	ALL
BERT	82.5	81.0	76.8	80.6	78.8	82.0	78.2	85.1	76.5	77.7	84.7	78.5	80.2
Fine-tuned BERT	86.9	84.1	81.7	84.4	84.2	86.7	80.2	89.2	82.0	79.8	88.6	81.5	84.1
PBLM-Max	83.3	82.5	77.6	84.2	77.6	82.5	71.4	87.8	80.4	79.8	87.1	74.2	80.7
HATN	85.4	83.5	78.8	82.2	78.0	81.2	80.0	87.4	83.2	81.0	85.9	81.2	82.3
PERL	89.9	85.0	**85.0**	86.5	87.0	89.9	**84.3**	90.6	87.1	84.6	90.7	81.9	86.9
R-PERL	**90.4**	**85.6**	84.8	**87.8**	**87.2**	**90.2**	83.9	**91.2**	**89.3**	**85.6**	**91.2**	**83.0**	**87.5**

	I → E	I → K	E → I	K → I	ALL	A → B	A → K	A → E	B → A	K → A	E → A	ALL	
BERT	75.4	78.8	72.2	70.6	74.2	70.9	78.8	77.1	72.1	74.0	81.0	75.6	
Fine-tuned BERT	81.5	78.0	77.6	78.7	78.9	72.9	81.9	83.0	79.5	76.3	82.8	79.4	
PBLM-Max	70.1	69.8	67.0	69.0	69.0	70.6	82.6	81.1	83.8	**87.4**	**87.7**	80.5	
HATN	74.0	74.4	74.8	78.9	75.5	58.7	68.8	64.1	77.6	78.5	83.0	71.8	
PERL	87.1	**86.3**	82.0	82.2	84.4	77.1	84.2	84.6	82.1	83.9	85.3	82.9	
R-PERL	**87.9**	86.0	**82.5**	**82.5**	**84.7**	**78.4**	**85.9**	**85.9**	**84.0**	85.1	85.9	**84.2**	

Table 1: Domain adaptation results. The top table is for the legacy product review domains of Blitzer et al. (2007) (denoted as the $P \Leftrightarrow P$ setups in the text). The bottom table involves selected legacy domains as well as the IMDb movie review domain (left; denoted as $P \Leftrightarrow I$) or the airline review domain (right; denoted as $P \Leftrightarrow A$). The All columns present averaged results across the setups to their left.

Num Sentences	BERT	Fine-tuned BERT	PERL	R-PERL
100	67.9	76.4	81.6	**83.9**
500	73.9	83.3	84.3	**84.6**
1K	75.3	83.9	84.6	**84.9**
2K	77.9	83.6	**85.3**	**85.3**
10K	80.9	86.9	87.1	**87.5**
20K	81.7	86.0	87.8	**88.1**

Table 2: In domain results on the IMDb movie review domain with increasing training set size.

an increased fraction of negative reviews (see §4). Finally, the IMDb and airline reviews are also more recent. The success of PERL in the $P \Leftrightarrow I$ and $P \Leftrightarrow A$ setups is of particular importance, as it indicates the potential of our algorithm to adapt supervised NLP algorithms to domains that substantially differ from their training domain.

Finally, our results clearly indicate the positive impact of a pivot-aware approach when fine-tuning BERT with unlabeled source and target data. Indeed, the averaged gaps between Fine-tuned BERT and BERT (3.9% for $P \Leftrightarrow P$, 4.7% for $P \Leftrightarrow I$, and 3.8% for $P \Leftrightarrow A$) are much smaller than the corresponding gaps between R-PERL and BERT (7.3% for $P \Leftrightarrow P$, 10.5% for $P \Leftrightarrow I$, and 8.6% for $P \Leftrightarrow A$).

In-domain Results In this setup both the labeled and the unlabeled data, used for supervised task training (labeled data, Step 3), fine-tuning (unlabeled data, Step 2), and pivot selection (both datasets) come from the same domain (IMDb). As shown in Table 2, PERL outperforms BERT and Fine-tuned BERT for all training set sizes.

Unsurprisingly, the impact of (R-)PERL diminishes as more labeled training data become available: From 7.5% (R-PERL vs. Fine-tuned BERT) when 100 sentences are available, to 2.1% for 20K training sentences. To our knowledge, the effectiveness of pivot-based methods for in-domain learning has not been demonstrated in the past.

6 Ablation Analysis and Discussion

In order to shed more light on PERL, we conduct an ablation analysis. We start by uncovering the hyper-parameters that have strong impact on its performance, and analyzing its stability across hyper-parameter configurations. We then explore the impact of some of the design choices we made when constructing the model.

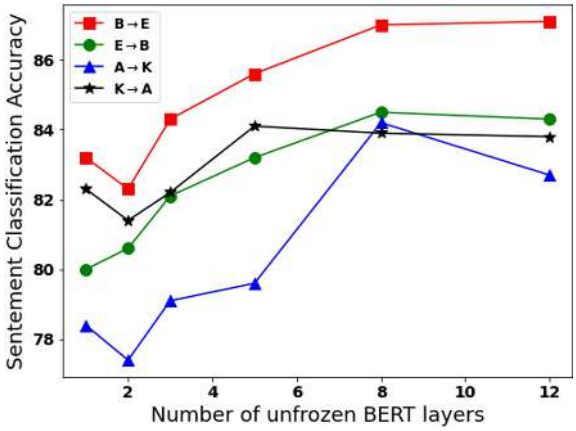

Figure 3: The impact of the number of unfrozen PERL layers during fine-tuning (Step 2).

In order to keep our analysis concise and to avoid heavy computations, we have to consider only a handful of arbitrarily chosen DA setups for each analysis. We follow the five-fold cross-validation protocol of §4 for hyper-parameter tuning, except that in some of the analyses a hyper-parameter of interest is kept fixed.

6.1 Hyper-parameter Analysis

In this analysis we focus on one hyper-parameter that is relevant only for methods that use massively pre-trained encoders (the number of unfrozen encoder layers during fine-tuning), as well as on two hyper-parameters that impact the core of our modified MLM objective (number of pivots and the pivot and non-pivot masking probabilities). We finally perform stability analysis across hyper-parameter configurations.

Number of Unfrozen BERT Layers during Fine Tuning (stage 2, Figure 1b) In Figure 3 we compare PERL final sentiment classification accuracy with six alternatives–1, 2, 3, 5, 8, or 12 unfrozen layers, going from the top to the bottom layers. We consider 4 arbitrarily chosen DA setups, where the number of unfrozen layers is kept fixed during the five-fold cross validation process. The general trend is clear: PERL performance improves as more layers are unfrozen, and this improvement saturates at 8 unfrozen layers (for the K→A setup the saturation is at 5 layers). The classification accuracy improvement (compared to 1 unfrozen layer) is of 4% or more in three of the setups (K→A is again

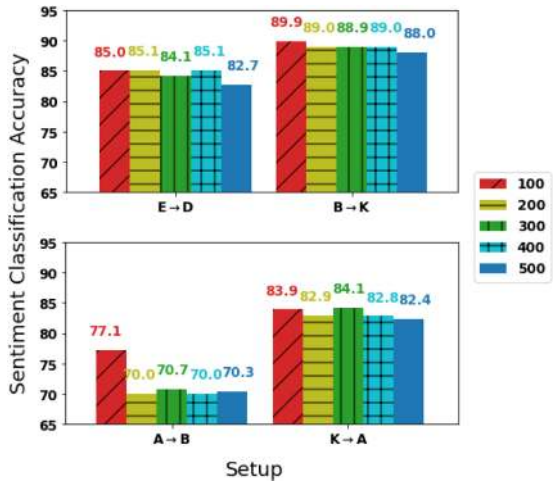

Figure 4: PERL sentiment classification accuracy across four setups with a varying number of pivots.

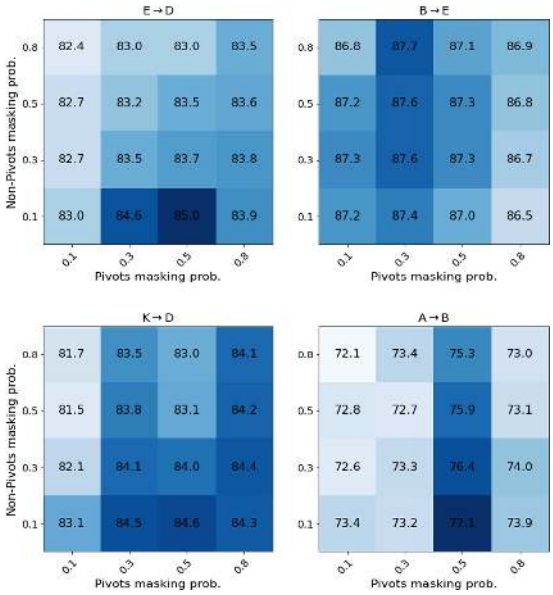

Figure 5: Heat maps of PERL performance with different pivot (α) and non-pivot (β) masking probabilities. A darker color corresponds to a higher sentiment classification accuracy.

the exception with only $\sim$ 2% improvement). Across the experiments of this paper, this hyper-parameter has been the single most influential hyper-parameter of the PERL, R-PERL and Fine-tuned BERT models.

Number of Pivots Following previous work (e.g., Ziser and Reichart, 2018), our hyper-parameter tuning process considers 100 to 500 pivots in steps of 100. We would next like to explore the impact of this hyper-parameter on PERL performance. Figure 4 presents our results, for four arbitrarily selected setups. In 3 of 4 setups PERL performance is stable across pivot numbers. In 2 setups, 100 is the optimal number of pivots (for the $A \rightarrow B$ setup with a large gap), and in the 2 other setups it lags behind the best value by no more than 0.2%. These two characteristics—model stability across pivot numbers and somewhat better performance when using fewer pivots—were observed across our experiments with PERL and R-PERL.

Pivot and Non-Pivot Masking Probabilities We next study the impact of the pivot and non-pivot masking probabilities, used during PERL fine-tuning (α and β, respectively, see §3). For both α and β we consider the values of 0.1, 0.3, 0.5, and 0.8. Figure 5 presents heat maps that summarize our results. A first observation is the relative stability of PERL to the values of these hyper-parameters: The gap between the best and worst performing configurations are 2.6% (E $\rightarrow$ D), 1.2% (B $\rightarrow$ E), 3.1% (K $\rightarrow$ D), and 5.0% (A $\rightarrow$ B). A second observation is that extreme α values (0.1 and 0.8) tend to harm the model. Finally, in 3 of 4 cases the best model performance is achieved with $\alpha = 0.5$ and $\beta = 0.1$.

Stability Analysis We finally turn to analyze the stability of the PERL models compared with the baselines. Previous work on PBLM and HATN has demonstrated their instability across model configurations (see Ziser and Reichart [2019] for PBLM and Cui et al. [2019] for HATN). As noted in Ziser and Reichart (2019), cross-configuration stability is of particular importance in unsupervised domain adaptation as the hyper-parameter configuration is selected using unlabeled data from the source, rather than the target domain.

In this analysis a hyper-parameter value is not considered for a model if it is not included in the best hyper-parameter configuration of that model for at least one DA setup. Hence, for PERL we fix the number of unfrozen layers (8), the number of pivots (100), and set $(\alpha, \beta) = (0.5, 0.1)$, and for PBLM we consider only word embedding size of 128 and 256. Other than that, we consider all possible hyper-parameter configurations of all models (§4, 54 configurations for PERL, R-PERL and Fine-tuned BERT, 18 for BERT, 30 for PBLM and 27 for HATN). Table 3 presents the minimum (min), maximum (max), average (avg), and standard deviation (std) of the test set scores

across the hyper-parameter configurations of each model, for 4 arbitrarily selected setups.

In all 4 setups, PERL and R-PERL consistently achieve higher avg, max, and min values and lower std values compared to the other models (with the exception of PBLM achieving higher max for $K \rightarrow A$). Moreover, the std values of PBLM and especially HATN are substantially higher than those of the models that use BERT. Yet, PERL and R-PERL demonstrate lower std values compared to BERT and Fine-tuned BERT in 3 of 4 setups, indicating that our method contributes to stability beyond the documented contribution of BERT itself Hao et al. (2019).

6.2 Design Choice Analysis

Impact of Pivot Selection One design choice that impacts our results is the method through which pivots are selected. We next compare three alternatives to our pivot selection method, keeping all other aspects of PERL fixed. As above, we arbitrarily select four setups.

We consider the following pivot selection methods: (a) Random-Frequent: Pivots are randomly selected from the unigrams and bigrams that appear at least 80 times in the unlabeled data of each of the domains; (b) High-MI, No Target: We select the pivots that have the highest mutual information (MI) with the source domain label, but appear less than 10 times in the target domain unlabeled data; (c) Oracle Miller (2019): Here the pivots are selected according to our method, but the labeled data used for pivot-label MI computation is the target domain test data rather than the source domain training data. This is an upper bound on the performance of our method since it uses target domain labeled data, which is not available to us. For all methods we select 100 pivots (see above).

Table 5 presents the results of the four PERL variants, and compare them to BERT and Fine-tuned BERT. We observe four patterns in the results. First, PERL with our pivot selection method, which emphasizes both high MI with the task label and high frequency in both the source and target domains, is the best performing model. Second, PERL with Random-Frequent pivot selection is substantially outperformed by PERL, but it still performs better than BERT (in 3 of 4 setups), probably because BERT is not tuned on unlabeled data from the participating domains. Yet, PERL with Random-Frequent pivots is outperformed by the Fine-tuned BERT in all setups, indicating that it provides a sub-optimal way of exploiting source and target unlabeled data. Third, in 3 of 4 setups, PERL with the High-MI, No Target pivots is outperformed by the baseline BERT model. This is a clear indication of the sub-optimality of this pivot selection method that yields a model that is inferior even to a model that was not tuned on source and target domain data. Finally, although, unsurprisingly, PERL with oracle pivots outperforms the standard PERL, the gap is smaller than 2% in all four cases. Our results clearly demonstrate the strong positive impact of our pivot selection method on the performance of PERL.

E→D				
	avg	**max**	**min**	**std**
R-PERL	84.6	85.8	83.1	0.7
PERL	85.2	86.0	84.4	0.4
Fine-tuned BERT	81.3	83.2	79.0	1.2
BERT	75.0	76.8	70.6	1.8
PBLM	71.7	79.3	65.9	3.4
HATN	73.7	81.1	53.9	10.7
B→K				
	avg	**max**	**min**	**std**
R-PERL	89.5	90.5	88.8	0.5
PERL	89.4	90.2	88.8	0.3
Fine-tuned BERT	86.9	87.7	84.9	0.8
BERT	81.1	82.5	78.6	1.1
PBLM	78.6	84.1	71.3	3.3
HATN	76.8	82.8	59.5	7.7
A→B				
	avg	**max**	**min**	**std**
R-PERL	75.3	79.0	72.0	1.7
PERL	73.9	77.1	70.9	1.7
Fine-tuned BERT	72.1	74.2	68.2	1.7
BERT	69.9	73.0	66.9	1.8
PBLM	64.2	71.6	60.9	2.7
HATN	57.6	65.0	53.7	3.5
K→A				
	avg	**max**	**min**	**std**
R-PERL	85.3	86.4	84.6	0.5
PERL	83.8	84.9	81.5	0.9
Fine-tuned BERT	77.8	82.1	67.1	4.2
BERT	70.4	74.0	65.1	2.6
PBLM	76.1	86.1	66.2	6.8
HATN	72.1	79.2	53.9	9.9

Table 3: Stability analysis.

	B → E				A → K			
	5 layers	8 layers	10 layers	12 layers (full)	5 layers	8 layers	10 layers	12 layers (full)
BERT	70.9	75.9	80.6	78.8	71.2	74.9	81.2	78.8
Fine-tuned BERT	74.6	76.5	84.2	84.2	74.0	76.3	80.8	81.9
PERL (Ours)	81.1	83.2	**88.2**	87.0	77.7	80.2	**84.7**	84.2

Table 4: Classification accuracy with reduced-size encoders.

	B → E	K → D	E → K	D → B
BERT	78.8	77.7	85.1	81.0
Fine-tuned BERT	84.2	79.8	89.2	84.1
High-MI, No Target	76.2	76.4	84.9	83.7
Random-Frequent	79.7	76.8	85.5	81.7
PERL (Ours)	**87.0**	**84.6**	**90.6**	**85.0**
Oracle	88.9	85.6	91.5	86.7

Table 5: Impact of PERL's pivot selection method.

	B → E	K → D	A → B	I → E
No fine-tuning				
BERT	78.8	77.7	70.9	75.4
Source data only				
Fine-tuned BERT	80.7	79.8	69.4	81.0
PERL	79.6	82.2	69.8	84.4
Target data only				
Fine-tuned BERT	82.0	80.9	71.6	81.1
PERL	86.9	83.0	71.8	84.2
Source and target data				
Fine-tuned BERT	84.2	79.8	72.9	81.5
PERL	**87.0**	**84.6**	**77.1**	**87.1**

Table 6: Impact of fine-tuning data selection.

Unlabeled Data Selection Another design choice we consider is the impact of the type of fine-tuning data. While we followed previous work (e.g., Ziser and Reichart, 2018) and used the unlabeled data from both the source and target domains, it might be that data from only one of the domains, particularly the target, is a better choice. As above, we explore this question on 4 arbitrarily selected domain pairs. The results, presented in Table 6, clearly indicate that our choice to use unlabeled data from both domains is optimal, particularly when transferring from a non-product domain (A or I) to a product domain.

Reduced Size Encoder We finally explore the effect of the fine-tuning step on the performance of reduced-size models. By doing this we address a major limitation of pre-trained encoders—their size, which prevents them from running on small computational devices and dictates long run times.

For this experiment we prune the top encoder layers before its fine-tuning step, yielding three new model sizes, with 5, 8, or 10 layers, compared with the full 12 layers. This is done both for Fine-tuned BERT and for PERL. We then tune the number of encoder's top unfrozen layers during fine-tuning, as follows: 5 layer-encoder (1, 2, 3); 8 layer-encoder (1, 3, 4, 5); 10 layer-encoder (1, 3, 5, 8); and full encoder (1, 2, 3, 5, 8, 12). For comparison, we utilize the BERT model when its top layers are pruned, and no fine-tuning is performed. We focus on two arbitrarily selected DA setups.

Table 4 presents accuracy results. In both setups PERL with 10 layers is the best performing model. Moreover, for each number of layers, PERL outperforms the other two models, with particularly substantial improvements for 5 and 8 layers (i.e., 7.3% and 6.7%, over BERT and

Fine-tuned BERT, respectively, for B → E and 8 layers).

Reduced-size PERL is of course much faster than the full model. The averaged run-time of the full (12 layers) PERL on our test-sets is 196.5 msec and 9.9 msec on CPU (skylake i9-7920X, 2.9 GHz, single thread) and GPU (GeForce GTX 1080 Ti), respectively. For 8 layers the numbers drop to 132.4 msec (CPU) and 6.9 msec (GPU) and for 5 layers to 84.0 (CPU) and 4.7 (GPU) msec.

7 Conclusions

We presented PERL, a domain-adaptation model that fine-tunes a massively pre-trained deep contextualized embedding encoder (BERT) with a pivot-based MLM objective. PERL outperforms strong baselines across 22 sentiment classification DA setups, improves in-domain model performance, increases its cross-configuration stability and yields effective reduced-size models.

Our focus in this paper is on binary sentiment classification, as was done in a large body of previous DA work. In future work we would like to extend PERL's reach to structured (e.g., dependency parsing and aspect-based sentiment classification) and generation (e.g., abstractive summarization and machine translation) NLP tasks.

Acknowledgments

We would like to thank the action editor and the reviewers, Yftah Ziser, as well as the members of the IE@Technion NLP group for their valuable feedback and advice. This research was partially funded by an ISF personal grant no. 1625/18.

References

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.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In John A. Carroll, Antal van den Bosch, and Annie Zaenen, editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics,

John Blitzer, Ryan T. McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In Dan Jurafsky and Éric Gaussier, editors, *EMNLP 2006, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 22-23 July 2006, Sydney, Australia*, pages 120–128. ACL.

Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. 2015. Unsupervised cross-domain word representation learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 730–740. The Association for Computer Linguistics.

Minmin Chen, Kilian Q. Weinberger, and Yixin Chen. 2011. Automatic feature decomposition for single view co-training. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 953–960. Omnipress.

Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress.

Stéphane Clinchant, Gabriela Csurka, and Boris Chidlovskii. 2016. A domain adaptation regularization for denoising autoencoders. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

Wanyun Cui, Guangyu Zheng, Zhiqiang Shen, Sihang Jiang, and Wei Wang. 2019. Transfer learning for sequences via learning to collocate.

In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500. Association for Computational Linguistics.

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. *J. Mach. Learn. Res.*, 17:59:1–59:35.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 513–520. Omnipress.

Stephan Gouws, Gert-Jan Van Rooyen, and Yoshua Bengio. 2012. Learning structural correspondences across different linguistic domains with synchronous neural language models. In *Proc. of the xLite Workshop on Cross-Lingual Technologies, NIPS*.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings: A case study in early modern english. *CoRR*, abs/1904.02817.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of BERT. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4141–4150. Association for Computational Linguistics.

Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. 2006. Correcting sample selection bias by unlabeled data. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann, editors, *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 601–608. MIT Press.

Hal Daumé III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *J. Artif. Intell. Res.*, 26:101–126.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In John A. Carroll, Antal van den Bosch, and Annie Zaenen, editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020,*

Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net.

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

Zheng Li, Ying Wei, Yu Zhang, and Qiang Yang. 2018. Hierarchical attention transfer network for cross-domain sentiment classification. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5852–5859. AAAI Press.

Zheng Li, Yu Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 2237–2243. ijcai.org.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard S. Zemel. 2016. The variational fair autoencoder. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

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 Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *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 142–150. The Association for Computer Linguistics.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2008. Domain adaptation with multiple sources. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1041–1048. Curran Associates, Inc.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 28–36. The Association for Computational Linguistics.

Timothy A. Miller. 2019. Simplified neural unsupervised domain adaptation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 414–419. Association for Computational Linguistics.

Quang Nguyen. 2015. The airline review dataset.

Thien Huu Nguyen, Barbara Plank, and Ralph Grishman. 2015. Semantic representations for domain adaptation: A case study on the tree kernel-based method for relation extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 635–644. The Association for Computer Linguistics.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 751–760. ACM.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *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 Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1498–1507. The Association for Computer Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language understandingpaper.pdf*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Brian Roark and Michiel Bacchiani. 2003. Supervised and unsupervised PCFG adaptation to novel domains. In Marti A. Hearst and Mari Ostendorf, editors, *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics.

Guy Rotman and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713.

Alexander M. Rush, Roi Reichart, Michael Collins, and Amir Globerson. 2012. Improved parsing and POS tagging using inter-sentence consistency constraints. In Jun'ichi Tsujii, James Henderson, and Marius Pasca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012*, Jeju Island, Korea, pages 1434–1444. ACL.

Tobias Schnabel and Hinrich Schütze. 2014. FLORS: fast and simple domain adaptation for part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 2:15–26.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. 2007. Direct importance estimation with model selection and its application to covariate shift adaptation. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 1433–1440. Curran Associates, Inc.

Manshu Tu and Bing Wang. 2019. Adding prior knowledge in hierarchical attention neural network for cross domain sentiment classification. *IEEE Access*, 7:32578–32588.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 1096–1103. ACM.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony

Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Gui-Rong Xue, Wenyuan Dai, Qiang Yang, and Yong Yu. 2008. Topic-bridged PLSA for cross-domain text classification. In Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong, editors, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 627–634. ACM.

Yi Yang and Jacob Eisenstein. 2014. Fast easy unsupervised domain adaptation with marginalized structured dropout. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 538–544. The Association for Computer Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 5754–5764.

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 236–246. The Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, pages 1441–1451. Association for Computational Linguistics.

Yftah Ziser and Roi Reichart. 2017. Neural structural correspondence learning for domain adaptation. In Roger Levy and Lucia Specia, editors, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 400–410. Association for Computational Linguistics.

Yftah Ziser and Roi Reichart. 2018. Pivot based language modeling for improved neural domain adaptation. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *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 1241–1251. Association for Computational Linguistics.

Yftah Ziser and Roi Reichart. 2019. Task refinement learning for improved accuracy and stability of unsupervised domain adaptation. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, pages 5895–5906. Association for Computational Linguistics.