Pivot Based Language Modeling for Improved Neural Domain Adaptation

Yftah Ziser and **Roi Reichart**
Faculty of Industrial Engineering and Management, Technion, IIT
syftah@campus.technion.ac.il, roiri@ie.technion.ac.il

Abstract

Representation learning with pivot-based methods and with Neural Networks (NNs) have lead to significant progress in domain adaptation for Natural Language Processing. However, most previous work that follows these approaches does not explicitly exploit the structure of the input text, and its output is most often a single representation vector for the entire text. In this paper we present the *Pivot Based Language Model (PBLM)*, a representation learning model that marries together pivot-based and NN modeling in a structure aware manner. Particularly, our model processes the information in the text with a sequential NN (LSTM) and its output consists of a context-dependent representation vector for every input word. Unlike most previous representation learning models in domain adaptation, PBLM can naturally feed structure aware text classifiers such as LSTM and CNN. We experiment with the task of cross-domain sentiment classification on 20 domain pairs and show substantial improvements over strong baselines.[1]

1 Introduction

Domain adaptation (*DA*, (Daumé III, 2007; Ben-David et al., 2010)) is a fundamental challenge in NLP, due to the reliance of many algorithms on costly labeled data which is scarce in many domains. To save annotation efforts, DA aims to import algorithms trained with labeled data from one or several domains to new ones. While DA algorithms have long been developed for many tasks and domains (e.g. (Jiang and Zhai, 2007; McClosky et al., 2010; Titov, 2011; Bollegala et al., 2011; Rush et al., 2012; Schnabel and Schütze, 2014)), the unprecedented growth of heterogeneous online content calls for more progress.

DA through Representation Learning (DReL), where the DA method induces shared representations for the examples in the source and the target domains, has become prominent in the Neural Network (NN) era. A seminal (non-NN) DReL work is structural correspondence learning (SCL) (Blitzer et al., 2006, 2007) which models the connections between pivot features – features that are frequent in the source and the target domains and are highly correlated with the task label in the source domain – and the other, non-pivot, features. While this approach explicitly models the correspondence between the source and the target domains, it has been outperformed by NN-based models, particularly those based on autoencoders (AEs, (Glorot et al., 2011; Chen et al., 2012)) which employ compress-based noise reduction to extract features that empirically support domain adaptation. Recently, Ziser and Reichart (2017) (ZR17) proposed to marry these approaches. They have presented the autoencoder-SCL models and demonstrated their superiority over a large number of previous approaches, particularly those that employ pivot-based ideas only or NNs only.

Current DReL methods, however, suffer from a fundamental limitation: they ignore the structure of their input text (usually sentence or document). This is reflected both in the way they represent their input text, typically with a single vector whose coordinates correspond to word counts or indicators across the text, and in their output which typically consists of a single vector representation. This structure-indifferent approach stands in a sharp contrast to numerous NLP algorithms where text structure plays a key role.

Moreover, learning a single feature vector per

[1] Our code is publicly available at: https://github.com/yftah89/PBLM-Domain-Adaptation.

Proceedings of NAACL-HLT 2018, pages 1241–1251
New Orleans, Louisiana, June 1 - 6, 2018.

input example, these methods can feed only task classifiers such as SVM and feed-forward NNs that take a single vector as input, but cannot feed sequential (e.g. RNNs and LSTMs (Hochreiter and Schmidhuber, 1997)) or convolution (CNNs (LeCun et al., 1998)) networks that require an input vector per word or sentence in their input. This may be a serious limitation given the excellent performance of structure aware models in a large variety of NLP tasks, including sentiment analysis and text classification (e.g.(Kim, 2014; Yogatama et al., 2017)) - prominent DA evaluation tasks.

Fig. 1 demonstrates the limitation of structure-indifferent modeling in DA for sentiment analysis. While the example review contains more positive pivot features (see definition in Sec. 2), the sentiment expressed in the review is negative. A representation learning method should encode the review structure (e.g. the role of the terms *at first* and *However*) in order to uncover the sentiment.[2]

In this paper we overcome these limitations. We present (Section 3) the *Pivot Based Language Model (PBLM)* - a domain adaptation model that (a) is aware of the structure of its input text; and (b) outputs a representation vector for every input word. Particularly, the model is a sequential NN (LSTM) that operates very similarly to LSTM language models (LSTM-LMs). The fundamental difference is that while for every input word LSTM-LMs output a hidden vector and a prediction of the next word, the output of PBLM is a hidden vector and a prediction of the next word if that word is a pivot feature or else, a generic NONE tag. Hence, PBLM not only exploits the sequential nature of its input text, but its output states can naturally feed LSTM and CNN task classifiers. Notice that PBLM is very flexible: instead of pivot based unigram prediction it can be defined to predict pivots of arbitrary length (e.g. the next bigram or trigram), or, alternatively, it can be defined over sentences or other textual units instead of words.

Following a large body of DA work, we experiment (Section 5) with the task of binary sentiment classification. We consider adaptation between each domain pair in the four product review domains of Blitzer et al. (2007) (12 domain pairs) as well as between these domains and an airline review domain (Nguyen, 2015) and vice versa (8 domain pairs). The latter 8 setups are particularly challenging as the airline reviews tend to be more negative than the product reviews (see Section 4).

[2]Pivots are defined with respect to a (source, target) domain pair. The pivots highlighted in the figure are the pivots for this review in all the setups we explored.

> I was *at first* **very excited** with my new Zyliss salad spinner - it is easy to spin and looks **great** ... . *However*, ... it doesn't get your greens very dry. I've been surprised and **disappointed** by the amount of water left on lettuce after spinning, and spinning, and spinning.

Figure 1: Example review from the kitchen appliances domain of Blitzer et al. (2007). Positive pivot features are underlined with a wavy line. Negative pivot features are underlined with a straight line. Although there are more positive pivots than negative ones, the review is negative.

We implement PBLM with two task classifiers, LSTM and CNN, and compare them to strong previous models, among which are: SCL (pivot based, no NN), the marginalized stacked denoising autoencoder model (MSDA, (Chen et al., 2012) - AE based, no pivots), the MSDA-DAN model ((Ganin et al., 2016) - AE with a Domain Adversarial Network (DAN) enhancement) and AE-SCL-SR (the best performing model of ZR17, combining AEs, pivot information and pre-trained word vectors). PBLM-LSTM and PBLM-CNN perform very similarly to each other and strongly outperform previous models. For example, PBLM-CNN achieves averaged accuracies of 80.4%, 84% and 76.2% in the 12 product domain setups, 4 product to airline setups and 4 airline to product setups, respectively, while AE-SCL-SR, the best baseline, achieves averaged accuracies of 78.1%, 78.7% and 68.1%, respectively.

2 Background and Previous Work

DA is an established challenge in machine learning in general and in NLP in particular (e.g. (Roark and Bacchiani, 2003; Chelba and Acero, 2004; Daumé III and Marcu, 2006)). While DA has several setups, the focus of this work is on unsupervised DA. In this setup we have access to unlabeled data from the the source and the target domains, but labeled data is available in the source domain only. We believe that in the current web era with the abundance of text from numerous domains, this is the most realistic setup.

Several approaches to DA have been proposed, for example: instance reweighting (Huang et al., 2007; Mansour et al., 2009), sub-sampling from

both domains (Chen et al., 2011) and learning joint target and source feature representations (DReL), the approach we take here. The rest of this section hence discusses DReL work that is relevant to our ideas, but first we describe our problem setup.

Unsupervised Domain Adaptation with DReL The pipeline of this setup typically consists of two steps: representation learning and classification. In the first step, a representation model is trained on the unlabeled data from the source and target domains. In the second step, a classifier for the supervised task is trained on the source domain labeled data. To facilitate domain adaptation, every example that is fed to the task classifier (second step) is first represented by the representation model of the first step. This is true both when the task classifier is trained and at test time when it is applied to the target domain.

An exception of this pipeline are end-to-end models that jointly learn to represent the data and to perform the classification task, exploiting the unlabeled and labeled data together. A representative member of this class of models (MSDA-DAN, (Ganin et al., 2016)) is one of our baselines.

Pivot Based Domain Adaptation This approach was proposed by Blitzer et al. (2006, 2007), through their SCL method. Its main idea is to divide the shared feature space of the source and the target domains to a set of pivot features that are frequent in both domains and have a strong impact on the source domain task classifier, and a complementary set of non-pivot features.

In SCL, after the original feature set is divided into the pivot and non-pivot subsets, this division is utilized in order to map the original feature space of both domains into a shared, low-dimensional, real-valued feature space. To do so, a binary classifier is defined for each of the pivot features. This classifier takes the non-pivot features of an input example as its representation, and is trained on the unlabeled data from both the source and the target domains, to predict whether its associated pivot feature appears in the example or not. Note that no human annotation is required for the training of these classifiers, the supervision signal is in the unlabeled data. The matrix whose columns are the weight vectors of the classifiers is post-processed with singular value decomposition (SVD) and the derived matrix maps feature vectors from the original space to the new.

Since the presentation of SCL, pivot-based DA has been researched extensively (e.g. (Pan et al., 2010; Gouws et al., 2012; Bollegala et al., 2015; Yu and Jiang, 2016; Ziser and Reichart, 2017)). PBLM is a pivot-based method but, in contrast to previous models, it relies on sequential NNs to exploit the structure of the input text. Even models such as (Bollegala et al., 2015), that embed pivots and non-pivots so that the former can predict if the latter appear in their neighborhood, learn a single representation for all the occurrences of a word in the input corpus. That is, Bollegala et al. (2015), as well as other methods that learn cross-domain word embeddings (Yang et al., 2017), learn word-type representations, rather than context specific representations. In Sec. 3 we show how PBLM's context specific outputs naturally feed structure aware task classifiers such as LSTM and CNN.

AE Based Domain Adaptation The basic elements of an autoencoder are an encoder function e and a decoder function d, and its output is a reconstruction of its input x: $r(x) = d(e(x))$. The parameters of the model are trained to minimize a loss between x and $r(x)$, such as their Kullback-Leibler (KL) divergence or their cross entropy.

Variants of AEs are prominent in recent DA literature. Examples include Stacked Denoising Autoencoders (SDA, (Vincent et al., 2008; Glorot et al., 2011) and marginalized SDA (MSDA, (Chen et al., 2012)) that is more computationally efficient and scalable to high-dimensional feature spaces than SDA, and has been extended in various manners (e.g. (Yang and Eisenstein, 2014; Clinchant et al., 2016)). Finally, models based on variational autoencoders (Kingma and Welling, 2014; Rezende et al., 2014) have recently been applied in DA (e.g. variational fair autoencoder (Louizos et al., 2016)), but in our experiments they were still not competitive with MSDA.

While AE based models have set a new state-of-the-art for DA in NLP, they are mostly based on noise reduction in the representation and do not exploit task specific and linguistic information. This paved the way for ZR17 that integrated pivot-based ideas into domain adaptation with AEs.

Combining Pivots and AEs in Domain Adaptation ZR17 combined AEs and pivot-based modeling for DA. Their basic model (AE-SCL) is a three layer feed-forward network where the non-pivot features are fed to the input layer, encoded

into a hidden representation and this hidden representation is then decoded into the pivot features of the input example. Their advanced model (AE-SCL-SR) has the same architecture but the decoding matrix consists of pre-trained embeddings of the pivot features, which encourages input documents with similar pivots to have similar hidden representations. These embeddings are induced by word2vec (Mikolov et al., 2013) trained with unlabeled data from the source and the target domains.

ZR17 have demonstrated the superiority of their models (especially, AE-SCL-SR) over SCL (pivot-based, no AE), MSDA (AE-based, no pivots) and MSDA-DAN (AE-based with adversarial enhancement, no pivots) in 16 cross-domain sentiment classification setups, including the 12 legacy setups of Blitzer et al. (2007). However, as in previous pivot based methods, AE-SCL and AE-SCL-SR learn a single, structure-indifferent, feature representation of the input text. Our core idea is to implement a pivot-based sequential neural model that exploits the structure of its input text and that its output representations can be smoothly integrated with structure aware classifiers such as LSTM and CNN. Our second goal is motivated by the strong performance of LSTM and CNN in text classification tasks (Yogatama et al., 2017).

3 Domain Adaptation with PBLMs

We now introduce our PBLM model that learns representations for DA. As PBLM is inspired by language modeling, we assume the original feature set of the NLP task classifier consists of word unigrams and bigrams. This choice of features also allows us to directly compare our work to the rich literature on DA for sentiment classification where this is the standard feature set. PBLM, however, is not limited to word n-gram features.

We start with a brief description of LSTM based language modeling (LSTM-LM, (Mikolov et al., 2010)) and then describe how PBLM modifies that model in order to learn pivot-based representations that are aware of the structure of the input text. We then show how to employ these representations in structure aware text classification (with LSTM or CNN) and how to train such PBLM-LSTM and PBLM-CNN classification pipelines.

LSTM Language Modeling LSTMs address the vanishing gradient problem commonly found in RNNs (Elman, 1990) by incorporating gating functions into their state dynamics (Hochreiter and Schmidhuber, 1997). At each time step, an LSTM maintains a hidden vector, h_t, computed in a sequence of non-linear transformations of the input x_t and the previous hidden states $h_1, \ldots, h_{t-1}$.

Given an input word, an LSTM-LM should predict the next word in the sequence. For a lexicon V, the probability of the j-th word is:

$$p(y_t = j) = \frac{e^{h_t \cdot W_j}}{\sum_{k=1}^{|V|} e^{h_t \cdot W_k}}$$

Here, W_i is a parameter vector learned by the network for each of the words in the vocabulary. The loss function we consider in this paper is the cross-entropy loss over these probabilities.

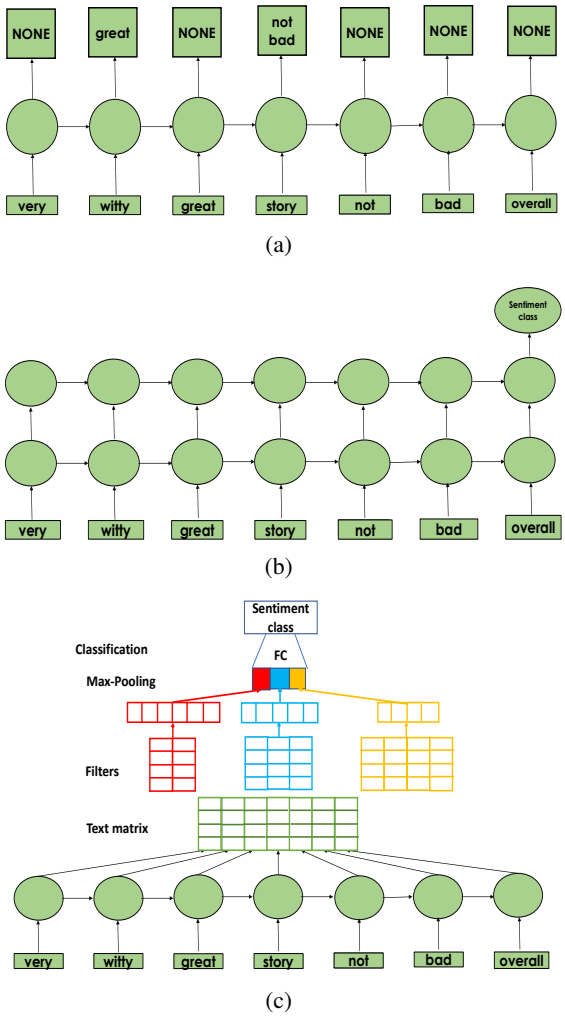

Figure 2: (a) Second order PBLM for representation learning. (b+c) PBLM based models for DA: PBLM-LSTM (b) and PBLM-CNN (c).

Representation Learning with PBLM Figure 2a provides an illustration of the PBLM model. The first (bottom) layer is an embedding layer,

where a 1-hot word vector input is multiplied by a (randomly initialized) parameter matrix before being passed to the next layer. The second layer is an LSTM that predicts the next bigram or unigram if one of these is a pivot (if both are, it predicts the bigram). Otherwise its prediction is NONE.

PBLM operates similarly to LSTM-LM. The basic difference between the models is the prediction they make for a given input word (x_t). While an LSTM-LM aims to predict the next input word, PBLM predicts the next word unigram or bigram if one of these is a pivot, and NONE otherwise.

PBLM is very flexible. It can be of any order: a k-order PBLM predicts the longest prefix of the sequence consisting of the next k words, as long as that prefix forms a pivot. If none of the prefixes forms a pivot then PBLM predicts NONE.[3] Moreover, while PBLM is defined here over word sequences, it can be defined over other sequences, e.g., the sentence sequence of a document.

Intuitively, in the example of fig. 2a a second order model is more informative for sentiment classification than a first-order model (that predicts only the next word unigram in case that word is a pivot) would be. Indeed, "not bad" conveys the relevant sentiment-related information, while "bad" is misleading with respect to that same sentiment. Notice that after the prefix "very witty" the model predicts "great" and not "great story" because in this example "great" is a pivot while "great story" is not, as "great story" is unlikely to be frequent outside the book review domain.

Figures 2a and 1 also demonstrate a major advantage of PBLM over models that learn a single text representation. From the book review example in fig. 2a, PBLM learns the connection between *witty* - an adjective that is often used to describe books, but not kitchen appliances - and *great* - a common positive adjective in both domains, and hence a pivot feature. Likewise, from the example of fig. 1 PBLM learns the connection between *easy* - an adjective that is often used to describe kitchen appliances, but not books - and *great*. That is, PBLM is able to learn the connection between *witty* and *easy* which will facilitate adaptation between the books and kitchen appliances domains. Previous work that learns a single text representation, in contrast, would learn from fig. 1 a connection between *easy* and the three pivots: *very excited*, *great* and *disappointed*. From fig. 2a such a method would learn the connection between *witty* and *great* and *not bad*. The connection between *witty* and *easy* will be much weaker.

Structure Aware Classification with PBLM Representations PBLM not only exploits the sequential nature of its input text, but its output vectors can feed LSTM (PBLM-LSTM, fig. 2b) and CNN (PBLM-CNN, fig. 2c) classifiers.

PBLM-LSTM is a three-layer model. The bottom two layers are the PBLM model of fig. 2a. When PBLM is combined with a classifier, its softmax layer (top layer of fig. 2a) is cut and only its output vectors (h_t) are passed to the next LSTM layer (third layer of fig. 2b). The final hidden vector of that layer feeds the task classifier.

Note that since we cut the PBLM softmax layer when it is combined with the task classifier, PBLM should be trained before this combination is performed. Below we describe how we exploit this modularity to facilitate domain adaptation.

In PBLM-CNN, the combination between the PBLM and the CNN is similar to fig. 2b: the PBLM's softmax layer is cut and a matrix whose columns are the h_t vectors of the PBLM is passed to the CNN. We employ K different filters of size $|h_t| \times d$, each going over the input matrix in a sliding window of d consecutive hidden vectors, and generating a $1 \times (n-d+1)$ size vector, where n is the input text length. A max pooling is performed for each of the k vectors to generate a single $1 \times K$ vector that is fed into the task classifier.

PBLM can feed structure aware classifiers other than LSTM and CNN. Moreover, PBLM can also generate a single text representation as in most previous work. This can be done, e.g., by averaging the PBLM's hidden vectors and feeding the averaged vector into a linear non-structured classifier (e.g. logistic regression) or a feed-forward NN. In Sec. 5 we demonstrate that PBLM's ability to feed structure aware classifiers such as LSTM and CNN provides substantial accuracy gains. To the best of our knowledge, PBLM is unique in its structure aware representation: previous work generated one representation per input example.

Domain Adaptation with PBLM Representations We focus on unsupervised DA where the input consists of a source domain labeled set and a plentiful of unlabeled examples form the source and the target domains. Our goal is to use the unlabeled data as a bridge between the domains.

[3] A word sequence is one of its own prefixes.

Our fundamental idea is to decouple the PBLM training which requires only unlabeled text, from the NLP classification task which is supervised and for which the required labeled example set is available only for the source domain. We hence employ a two step training procedure. First PBLM (figure 2a) is trained with unlabeled data from both the source and the target domains. Then the trained PBLM is combined with the classifier layers (top layer of fig. 2b, CNN layers of fig. 2c) and the final model is trained with the source domain labeled data to perform the classification task. As noted above, in the second step we cut the PBLM's softmax layer, only its h_t vectors are passed to the classifier. Moreover, during this step the parameters of the pre-trained PBLM are held fixed, only the parameters of the classifier layers are trained.

4 Experimental Setup

[4]**Task and Domains** Following a large body of DA work, we experiment with the task of cross-domain sentiment classification. To facilitate comparison with previous work we experiment with the product review domains of (Blitzer et al., 2007) – Books (B), DVDs (D), Electronic items (E) and Kitchen appliances (K) (12 ordered domain pairs) – replicating the experimental setup of ZR17 (including baselines, design, and hyper-parameter details). For each domain there are 2000 labeled reviews, 1000 positive and 1000 negative, and unlabeled reviews: 6000 (B), 34741 (D), 13153 (E) and 16785 (K).

To consider a more challenging setup we experiment with a domain consisting of user reviews on services rather than products. We downloaded an airline review dataset, consisting of reviews labeled by their authors (Nguyen, 2015). We randomly sampled 1000 positive and 1000 negative reviews for our labeled set, the remaining 39396 reviews form our unlabeled set. We hence have 4 product to airline and 4 airline to product setups.

Interestingly, in the product domains unlabeled reviews tend to be much more positive than in the airline domain. Particularly, in the B domain there are 6.43 positive reviews on every negative review; in D the ratio is 7.39 to 1; in E it is 3.65 to 1; and in K it is 4.61 to 1. In the airline domain there are only 1.15 positive reviews for every negative review. We hence expect DA from product to airline reviews and vice versa to be more challenging than DA from one product review domain to another.[5]

Baselines We consider the following baselines: (a) AE-SCL-SR (ZR17). We also experimented with the more basic AE-SCL but, like in ZR17, we got lower results in most cases; (b) SCL with pivot features selected using the mutual information criterion (SCL-MI, (Blitzer et al., 2007)). For this method we used the implementation of ZR17; (c) MSDA (Chen et al., 2012), with code taken from the authors' web page; (d) The MSDA-DAN model (Ganin et al., 2016) which employs a domain adversarial network (DAN) with the MSDA vectors as input. The DAN code is taken from the authors' repository; (e) The no domain adaptation case where the sentiment classifier is trained in the source domain and applied to the target domain without adaptation. For this case we consider three classifiers: logistic regression (denoted NoSt as it is not aware of its input's structure), as well as LSTM and CNN which provide a control for the importance of the structure aware task classifiers in PBLM models. To further control for this effect we compare to the PBLM-NoSt model where the PBLM output vectors (h_t vectors generated after each input word) are averaged and the averaged vector feeds the logistic regression classifier.[6]

In all the participating methods, the input features consist of word unigrams and bigrams. The division of the feature set into pivots and non-pivots is based on the the method of ZR17 that followed the work of Blitzer et al. (2007) (details are in Appendix C). The sentiment classifier employed with the SCL-MI, MSDA and AE-SCL-SR representations is the same logistic regression classifier as in the NoSt condition mentioned above. For these methods we concatenate the representation learned by the model with the original representation and this representation is fed to the classifier. MSDA-DAN jointly learns the feature representation and performs the sentiment classification task. It is hence fed by a concatenation of the original and the MSDA-induced representations.

[4]The URLs of the datasets and the code (previous models and standard packages) we used, are in Appendix A.

[5]While we have the labels for our unlabeled data, we did not use them in our research except in this analysis.

[6]We considered several additional baselines: (1) Variational fair autoencoder (Louizos et al., 2016) which performed substantially worse than the DA baselines ((a)-(d)); (2) We tried to compare to (Bollegala et al., 2015) but, similarly to ZR17, failed to replicate their results; and (3) We replaced PBLM with an LSTM-LM, but the results substantially degraded. We do not report results for these models.

Five Fold CV We employ a 5-fold cross-validation protocol as in (Blitzer et al., 2007; Ziser and Reichart, 2017). In all five folds 1600 source domain examples are randomly selected for training data and 400 for development, such that both the training and the development sets are balanced and have the same number of positive and negative reviews. The results we report are the averaged performance of each model across these 5 folds.

Hyperparameter Tuning For all previous models, we follow the tuning process described in ZR17 (paper and appendices). Hyperparameter tuning for the PBLM models and the non-adapted CNN and LSTM is described in Appendix B.

5 Results

Overall Results Table 1 presents our results. PBLM models with structure aware classifiers (PBLM-LSTM and PBLM-CNN, henceforth denoted together as S-PBLM) outperform all other alternatives in all 20 setups and three averaged evaluations (*All* columns in the tables). The gaps are quite substantial – the average accuracy of PBLM-LSTM and PBLM-CNN compared to the best baseline, AE-SCL-SR, are: 79.6% and 80.4% vs. 78.1% for the product review setups, 85% and 84% vs. 78.7% for the product to airline (service) review setups, and 76.1% and 76.2% vs. 68.1% for the airline to product review setups.

S-PBLM performance in the more challenging product to airline and airline to product setups are particularly impressive. The challenging nature of these setups stems from the presumably larger differences between product and service reviews and from the different distribution of positive and negative reviews in the unlabeled data of both domains (Sec. 4). These differences are reflected by the lower performance of the non-adapted classifiers: an averaged accuracy of 70.6%-73.1% across product domain pairs (three lower lines of the *All* column of the top table), compared to an average of 67.3%-69.9% across product to airline setups and an average of 61.3%-62.4% across airline to product setups. Moreover, while the best previous method (AE-SCL-SR) achieves an averaged accuracy of 78.1% for product domains and an averaged accuracy of 78.7% when adapting from product to airline reviews, when adapting from airline to product reviews its averaged accuracy drops to 68.1%. The S-PBLM models do consistently better in all three setups, with an averaged accuracy of 80.4%, 85% and 76.2% of the best S-PBLM model, respectively.

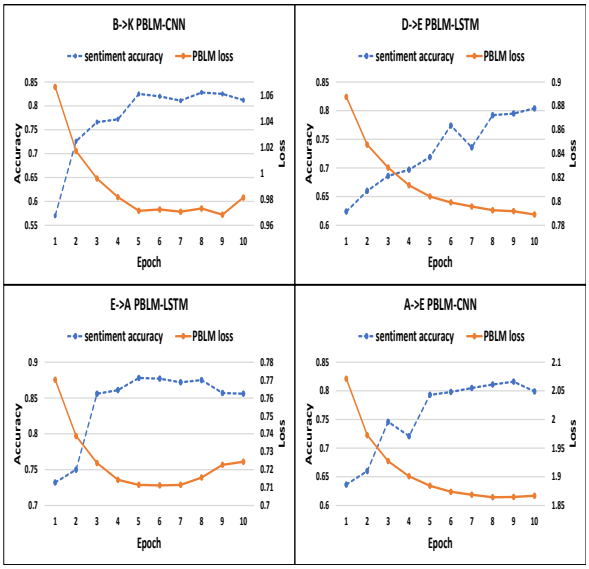

Figure 3: PBLM loss (solid, red line) vs. sentiment accuracy (dashed, blue line) of PBLM-CNN (top) and PBLM-LSTM (bottom) in four representative setups. Patterns in other setups are very similar.

Analysis of S-PBLM Strength The results shed light on the sources of the S-PBLM models success. The accuracy of these models, PBLM-LSTM and PBLM-CNN, is quite similar across setups: their accuracy gap is up to 3.1% in all 20 setups and up to 1% in the three averages (*All* columns). However, the S-PBLM models substantially outperform PBLM-NoSt that employs a structure-indifferent classifier. The averaged gaps are 5.6% (80.4% vs. 74.8%) in the product to product setups, 11.1% in the product to airline setups (85% vs. 73.9%) and 10.9% in the airline to product setups (76.2% vs. 65.3%). Hence, we can safely conclude that while the integration of PBLM with a structured task classifier has a dramatic impact on cross-domain accuracy, it is less important if that classifier is an LSTM or a CNN.

Comparison with non-adapted models reveals that structure aware modeling, as provided by LSTM and CNN, is not sufficient for high performance. Indeed, non-adapted LSTM and CNN do substantially worse than S-PBLM in all setups. Finally, comparison with AE-SCL-SR demonstrates that while the integration of pivot based learning with NNs leads to stronger results than in any other previous work, the structure awareness of the S-PBLM models substantially improves accuracy.

Product Review Domains (Blitzer et al., 2007)													
Source-Target	**D-B**	**E-B**	**K-B**	**B-D**	**E-D**	**K-D**	**B-E**	**D-E**	**K-E**	**B-K**	**D-K**	**E-K**	**All**
PBLM Models													
PBLM-LSTM	80.5	70.8	73.5	82.6	**77.6**	78.6	74.5	**80.4**	85.4	80.9	**83.3**	87.1	79.6
PBLM-CNN	**82.5**	**71.4**	**74.2**	**84.2**	75	**79.8**	**77.6**	79.6	**87.1**	**82.5**	83.2	**87.8**	**80.4**
PBLM-NoSt	74	68.6	67.4	78.3	73.2	73.3	71.3	74.2	82.1	75.5	76.9	83.2	74.8
Previous Work Models													
AE-SCL-SR	77.3	71.1	73	81.1	74.5	76.3	76.8	78.1	84	80.1	80.3	84.6	78.1
MSDA	76.1	71.9	70	78.3	71	71.4	74.6	75	82.4	78.8	77.4	84.5	75.9
MSDA-DAN	75	71	71.2	79.7	73.1	73.8	74.7	74.5	82.1	75.4	77.6	85	76.1
SCL-MI	73.2	68.5	69.3	78.8	70.4	72.2	71.9	71.5	82.2	77.2	74	82.9	74.3
No Domain Adaptation													
NoSt	73.6	67.9	67.6	76	69.1	70.2	70	70.9	81.6	74	73.2	82.4	73.1
LSTM	69.2	67.9	67.5	72.8	68.1	66.2	65.9	68.3	78.2	72.1	70.5	80.6	70.6
CNN	71.2	65.6	66.5	73.6	67.1	70.8	69.6	69.7	79.9	72.7	72.6	80.6	71.6

Product and Airline Review Domains (Blitzer et al., 2007; Nguyen, 2015)										
Source-Target	B-A	D-A	E-A	K-A	All (P-Air)	A-B	A-D	A-E	A-K	All (Air-P)
PBLM Models										
PBLM-LSTM	83.7	**81**	**87.7**	**87.4**	**85**	70.3	71.1	80.5	**82.6**	76.1
PBLM-CNN	**83.8**	78.3	86.5	86.1	84	**70.6**	**71.3**	**81.1**	81.8	**76.2**
PBLM-NoSt	74.2	74.9	72.4	73.9	73.9	62.5	62	69.6	67.3	65.3
Previous Work Models										
AE-SCL-SR	79.1	76.1	82.6	76.9	78.7	60.5	66	74.4	71.7	68.1
MSDA	72.2	73.3	75.1	76.8	74.3	58.5	61	70.6	69	64.8
MSDA-DAN	73.5	73.9	76.3	76.6	75	59.5	60.7	71	71.7	65.7
SCL	70.9	69	80.2	72.3	73	61.7	62.1	72.3	69.7	66.4
No Domain Adaptation										
NoSt	68.5	67.6	74	69.6	69.9	57.5	59.7	67.2	65.2	62.4
LSTM	68.3	65	72.1	68.6	67.3	56.7	57.3	66.2	65	61.3
CNN	67.6	66.7	72	70	69.1	56.3	59	66	66.6	62

Table 1: Accuracy of adaption between product review domains (top table). and between product review domains and the airline (A) review domain (bottom table). All the differences between PBLM-CNN and AE-SCL-SR and between PBLM-LSTM and AE-SCL-SR are statistically significant (except from E-B in the former comparison and E-B and K-B in the latter). Statistical significance is computed with the McNemar paired test for labeling disagreements ((Gillick and Cox, 1989; Blitzer et al., 2006), $p < 0.05$).

Figure 3 further demonstrates the adequacy of the PBLM architecture for domain adaptation. The graphs demonstrate, for both S-PBLM models, a strong correlation between the PBLM cross-entropy loss values and the sentiment accuracy of the resulting PBLM-LSTM and PBLM-CNN models. We show these patterns for two product domain setups and two setups that involve a product domain and the airline domain – the patterns for the other setups of table 1 are very similar.

This analysis highlights our major contribution. We have demonstrated that it is the combination of four components that makes DA for sentiment classification very effective: (a) Neural network modeling; (b) Pivot based modeling; (c) Structure awareness of the pivot-based model; and (d) Structure awareness of the task classifier.

6 Conclusions

We addressed the task of DA in NLP and presented PBLM: a representation learning model that combines pivot-based ideas and NN modeling, in a structure aware manner. Unlike previous work, PBLM exploits the structure of its input, and its output consists of a vector per input word. PBLM-LSTM and PBLM-CNN substantially outperform strong previous models in traditional and newly presented sentiment classification DA setups.

In future we intend to extend PBLM so that it could deal with NLP tasks that require the prediction of a linguistic structure. For example, we believe that PBLM can be smoothly integrated with recent LSTM-based parsers (e.g. (Dyer et al., 2015; Kiperwasser and Goldberg, 2016; Dozat and Manning, 2017)). We also intend to extend the reach of our approach to cross-lingual setups.

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. https://doi.org/10.1007/s10994-009-5152-4.

John Blitzer, Mark Dredze, Fernando Pereira, et al. 2007. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proc. of ACL*. http://aclweb.org/anthology/P07-1056.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proc. of EMNLP*. http://aclweb.org/anthology/W06-1615.

Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. 2015. Unsupervised cross-domain word representation learning. In *Proc. of ACL*. https://doi.org/10.3115/v1/P15-1071.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2011. Relation adaptation: learning to extract novel relations with minimum supervision. In *Proc. of IJCAI*. https://doi.org/10.1109/TKDE.2011.250.

Ciprian Chelba and Alex Acero. 2004. Adaptation of maximum entropy capitalizer: Little data can help a lot. In *Proc. of EMNLP*. http://aclweb.org/anthology/W04-3237.

Minmin Chen, Yixin Chen, and Kilian Q Weinberger. 2011. Automatic feature decomposition for single view co-training. In *Proc. of ICML*. http://dblp.uni-trier.de/rec/bib/conf/icml/ChenWC11.

Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proc. of ICML*. http://icml.cc/2012/papers/416.pdf.

Stéphane Clinchant, Gabriela Csurka, and Boris Chidlovskii. 2016. A domain adaptation regularization for denoising autoencoders. In *Proc. of ACL (short papers)*. https://doi.org/10.18653/v1/P16-2005.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proc. of ACL*. http://aclweb.org/anthology/P07-1009.

Hal Daumé III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research* 26:101–126. http://dl.acm.org/citation.cfm?id=1622559.1622562.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proc. of ICLR*.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proc. of ACL*. http://www.aclweb.org/anthology/P15-1033.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14(2):179–211.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17(59):1–35. http://jmlr.org/papers/v17/15-239.html.

Laurence Gillick and Stephen J Cox. 1989. Some statistical issues in the comparison of speech recognition algorithms. In *Proc. of ICASSP*. IEEE. https://doi.org/10.1109/ICASSP.1989.266481.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *In proc. of ICML*. pages 513–520. http://dblp.uni-trier.de/rec/bib/conf/icml/GlorotBB11.

Stephan Gouws, GJ Van Rooyen, MIH Medialab, 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*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. 2007. Correcting sample selection bias by unlabeled data. In *Proc. of NIPS*.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp. In *Proc. of ACL*. http://aclweb.org/anthology/P07-1034.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *In Proc. of EMNLP*. http://www.aclweb.org/anthology/D14-1181.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proc. of ICLR*. http://dblp.uni-trier.de/rec/bib/journals/corr/KingmaW13.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the ACL (TACL)* 4:313–327. https://transacl.org/ojs/index.php/tacl/article/view/885.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324. https://doi.org/10.1109/5.726791.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. 2016. The variational fair autoencoder http://dblp.uni-trier.de/rec/bib/journals/corr/LouizosSLWZ15.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain adaptation with multiple sources. In *Proc. of NIPS*.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Proc. of NAACL*. http://aclweb.org/anthology/N10-1004.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*. https://doi.org/10.1109/AINL-ISMW-FRUCT.2015.7382966.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proc. of NIPS*.

Quang Nguyen. 2015. The airline review dataset. `https://github.com/quankiquanki/skytrax-reviews-dataset`. Scraped from `www.airlinequality.com`.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*. ACM, pages 751–760. https://doi.org/10.1145/1772690.1772767.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proc. of ICML*. http://dblp.uni-trier.de/rec/bib/conf/icml/RezendeMW14.

Brian Roark and Michiel Bacchiani. 2003. Supervised and unsupervised pcfg adaptation to novel domains. In *Proc. of HLT-NAACL*. http://aclweb.org/anthology/N03-1027.

Alexander M Rush, Roi Reichart, Michael Collins, and Amir Globerson. 2012. Improved parsing and pos tagging using inter-sentence consistency constraints. In *Proc. of EMNLP-CoNLL*. http://aclweb.org/anthology/D12-1131.

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. http://aclweb.org/anthology/Q/Q14/Q14-1002.pdf.

Ivan Titov. 2011. Domain adaptation by constraining inter-domain variability of latent feature representation. In *Proc. of ACL*. http://www.aclweb.org/anthology/P11-1007.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proc. of ICML*. https://doi.org/10.1145/1390156.1390294.

Wei Yang, Wei Lu, and Vincent Zheng. 2017. A simple regularization-based algorithm for learning cross-domain word embeddings. In *Proc. of EMNLP*. https://www.aclweb.org/anthology/D17-1312.

Yi Yang and Jacob Eisenstein. 2014. Fast easy unsupervised domain adaptation with marginalized structured dropout. In *Proc. of ACL (short papers)*. https://doi.org/10.3115/v1/P14-2088.

Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom. 2017. Generative and discriminative text classification with recurrent neural networks. *arXiv preprint arXiv:1703.01898* .

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proc. of EMNLP*. http://aclweb.org/anthology/D16-1023.

Yftah Ziser and Roi Reichart. 2017. Neural structural correspondence learning for domain adaptation. In *Proc. of CoNLL*. http://aclweb.org/anthology/K17-1040.

A URLs of Code and Data

As mentioned in section 4 of the paper, we provide here a list of URLs for the code and data we use in the paper. We do that in order to avoid a large number of footnotes in the main paper:

- Blitzer et al. (2007) product review data: `http://www.cs.jhu.edu/~mdredze/datasets/sentiment/index2.html`.
- The airline review data is (Nguyen, 2015).
- Code for the AE-SCL and AE-SCL-SR models of ZR17 (Ziser and Reichart, 2017): `https://github.com/yftah89/Neural-SCLDomain-Adaptation`.
- Code for the SCL-MI method of Blitzer et al. (2007): see footnote [7] (the URL does not fit into the line width).

[7] `https://github.com/yftah89/structural-correspondence-learning-SCL`

- Code for MSDA (Chen et al., 2012): http://www.cse.wustl.edu/~mchen.
- Code for the domain adversarial network used as part of the MSDA-DAN baseline (Ganin et al., 2016): https://github.com/GRAAL-Research/domain_adversarial_neural_network.
- Logistic regression code: http://scikit-learn.org/stable/.

B Hyperparameter Tuning and Experimental Details

Hyperparameter Tuning As discussed in section 4 of the paper, for all previous work models, we follow the experimental setup of ZR17 (paper and appendices) including their hyperparameter estimation protocol. The hyperparameters of the PBLM models and the non-adapted CNN and LSTM are provided here. For PBLM we considered the following hyperparameteres:

- Input word embedding size: $(32, 64, 128, 256)$.
- Number of pivot features: $(100, 200, 300, 400, 500)$.
- $|h_t|$: $(128, 256, 512)$.
- PBLM model order: second order.

For the LSTM in PBLM-LSTM as well as the baseline non-adapted LSTM we considered the same $|h_t|$ and input word embedding size values as for PBLM. For PBLM-CNN and for the baseline, non-adapted, CNN we only experimented with $K = 250$ filters and with a kernel of size $d = 3$.

All the algorithms in the paper that involve a CNN or a LSTM (including the PBLM itself) are trained with the ADAM algorithm (Kingma and Ba, 2015). For this algorithm we used the parameters described in the original ADAM article:

- Learning rate: $lr = 0.001$.
- Exponential decay rate for the 1st moment estimates: $\beta_1 = 0.9$.
- Exponential decay rate for the 2nd moment estimates: $\beta_2 = 0.999$.
- Fuzz factor: $\epsilon = 1e - 08$.
- Learning rate decay over each update: $decay = 0.0$.

Experimental Details All sequential models considered in our experiments are fed with one review example at a time. For all models in the paper, punctuation is first removed from the text before it is processed by the model (sentence boundaries are still encoded). This is the only preprecessing step we employ in the paper.

We considered several alternative implementations of the PBLM-NoSt baseline. In the variant we selected the PBLM output vectors (h_t vectors generated after each word of the input review) are averaged and the averaged vector feeds a non-structured logistic regression classifier. We also tried to take only the final h_t vector of PBLM as an input to the classifier or to sum the h_t vectors instead of taking their average. These alternatives gave worse results.

C Pivot Feature Selection

As mentioned in the main paper, the division of the feature set into pivots and non-pivots is based on the unlabeled data from both the source and the target domains, using the method of ZR17 (which is in turn based on (Blitzer et al., 2007)). Here we provide the details of the pivot selection criterion.

Pivot features are frequent in the unlabeled data of both the source and the target domains, appearing at least 10 times in each, and among those features are the ones with the highest mutual information with the task (sentiment) label in the source domain labeled data. For non-pivot features we consider unigrams and bigrams that appear at least 10 times in their domain.