Deep Pivot-Based Modeling for Cross-language Cross-domain Transfer with Minimal Guidance

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

Abstract

While cross-domain and cross-language transfer have long been prominent topics in NLP research, their combination has hardly been explored. In this work we consider this problem, and propose a framework that builds on pivot-based learning, structure-aware Deep Neural Networks (particularly LSTMs and CNNs) and bilingual word embeddings, with the goal of training a model on labeled data from one (language, domain) pair so that it can be effectively applied to another (language, domain) pair. We consider two setups, differing with respect to the unlabeled data available for model training. In the *full* setup the model has access to unlabeled data from both pairs, while in the *lazy* setup, which is more realistic for truly resource-poor languages, unlabeled data is available for both domains but only for the source language. We design our model for the lazy setup so that for a given target domain, it can train once on the source language and then be applied to any target language without re-training. In experiments with nine English-German and nine English-French domain pairs our best model substantially outperforms previous models even when it is trained in the lazy setup and previous models are trained in the full setup.[1]

1 Introduction

The field of Natural Language Processing (NLP) has made impressive progress in the last two decades and text processing applications are now performed in a quality that was beyond imagination only a few years ago. With this success, it is only natural that researchers seek ways to apply NLP algorithms in as many languages and textual domains as possible. However, the success of NLP algorithms most often relies on the availability of non-trivial supervision such as corpora annotated with linguistic classes or structures, and for multilingual applications often also on parallel corpora. This resource bottleneck seriously challenges the world-wide accessibility of NLP technology.

To address this problem substantial efforts have been put into the development of cross-domain (*CD*, (Daumé III, 2007; Ben-David et al., 2010)) and cross-language (CL) transfer methods. For both areas, while a variety of methods have been developed for many tasks throughout the years (§ 2), with the prominence of deep neural networks (DNNs) the focus of modern methods is shifting towards learning data representations that can serve as a bridge across domains and languages.

For CD, this includes: (a) pre-DNN work ((Blitzer et al., 2006, 2007), known as *structural correspondence learning (SCL)*), that 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; (b) DNN work (Glorot et al., 2011; Chen et al., 2012) which employs compress-based noise reduction to learn cross-domain features; and recently also (c) works that combine the two approaches (Ziser and Reichart, 2017, 2018) (henceforth ZR17 and ZR18). For CL, the picture is similar: multilingual representations (usually word embeddings) are prominent in the transfer of NLP algorithms from one language to another (e.g. (Upadhyay et al., 2016)).

In this paper we aim to take CL and CD transfer a significant step forward and design methods that can adapt NLP algorithms simultaneously across languages and domains. We consider this research problem fundamental to our field as manually annotated resources are often scarce in many domains, even for languages that are consid-

[1] Our code is publicly available at https://github.com/yftah89/PBLM-Cross-language-Cross-domain

ered resource-rich. With effective cross-language cross-domain (CLCD) methods it is sufficient to have training resources in a single domain of one language in order to solve the task in any other (language, domain) pair.

As a first step, our focus in this work is on the task of sentiment classification that has been extensively researched in the CD literature. Surprisingly, even for this task we are aware of only one previous work that aims to perform CLCD learning (Fernández et al., 2016). However, this work does not employ modern DNN techniques and is substantially outperformed by our methods.

Our approach to CLCD learning is rooted in the family of methods that combine the power of both DNNs and pivot-based ideas, and is based on two principles. First, we build on the recent progress in learning multilingual word embeddings (Ruder et al., 2017). Such embeddings help close the lexical gap between languages as they map their different vocabularies to a shared vector space. Second, we follow (Prettenhofer and Stein, 2010, 2011; Fernández et al., 2016) and redefine the concept of pivot features for CLCD setups (§ 5). While these authors already employed this idea in order to design pivot-based methods in CL (Prettenhofer and Stein, 2010, 2011) and CLCD (Fernández et al., 2016) for text classification and sentiment analysis, their algorithms do not employ DNNs and multilingual embeddings. In this paper we show that it is the combination of bilingual word embeddings *(BEs)* and structure aware DNNs with the re-defined pivots that leads to high quality CLCD models.

Aiming to facilitate transfer to resource poor languages and domains, our methods rely on as little supervision as possible. Particularly, we explore two scenarios. In the first, *full CLCD* setup, models have access to manually annotated reviews from the source (language, domain) pair, and unannotated reviews from both the source and the target (language, domain) pairs. In the second, *lazy CLCD* setup, models have access only to source language reviews - annotated reviews from the source domain, and unannotated reviews from both the source and the target domains.

We consider the lazy setup to be the desired standard setup of CLCD learning for two reasons. First, in true resource-poor languages we expect it to be hard to find a sufficient number of reviews from many domains, even if they are unannotated (imagine for example trying to obtain 50K unlabeled spinner reviews in Swahili). Second, it allows a *train once, adapt everywhere* mode: instead of training a separate model for each target language, in this setup for each target domain only a single model is trained on the source language, and the target language is considered only at test time through BEs (§ 5). Notice that in order to allow the lazy setup, the BEs should be trained such that the source language embeddings have no knowledge about any particular target language. In § 5 we discuss the BEs we employ (Smith et al., 2017), which have this property.

We create CLCD variants of DNN- and pivot-based methods originally designed to learn effective representations for CD learning. To the best of our knowledge, there are three such methods, which employ two types of DNNs (§ 4): (a) AE-SCL and AE-SCL-SR (Ziser and Reichart, 2017) that integrate pivot-based ideas (SCL) with autoencoder-based (AE) noise reduction; and (b) pivot-based language modeling (PBLM, (Ziser and Reichart, 2018)) that combines pivot-based ideas with LSTMs for representation learning, and integrates this architecture with an LSTM or a CNN for task classification. In § 5 we discuss how to employ these methods for CLCD transfer where the lexical gap between languages is bridged by pivot translation and BEs, and show that PBLM allows for more effective transfer.

We address the task of binary sentiment classification and experiment with nine English-German and nine English-French domain pairs (§ 6, 7). Our PBLM-based models substantially outperform all previous models, even when the PBLM model is trained in the lazy setup and the previous models are trained in the full setup.

2 Previous Work

We briefly survey work on CL and CD learning and on multilingual word embeddings. We focus on aspects that are relevant to our work rather than on a comprehensive survey of the extensive previous work on these problems.

Cross-language transfer CL has been explored extensively in NLP. Example applications include POS tagging (Täckström et al., 2013), syntactic parsing (Guo et al., 2015; Ammar et al., 2016), text classification (Shi et al., 2010; Prettenhofer and Stein, 2010) and sentiment analysis (Wan, 2009; Zhou et al., 2016) among others.

Our work is mostly related to two works: (a) Cross-lingual SCL (CL-SCL, (Prettenhofer and Stein, 2010, 2011)); and (b) Distributional Correspondence Indexing (DCI, (Fernández et al., 2016)) – in both cases pivot features were redefined to support CL (in (a)) and CLCD (in (b)) with non-DNN models, in order to perform sentiment analysis. Below we show how we combine this idea with modern DNNs and BEs to substantially improve CLCD learning.

Cross-domain transfer In NLP, CD transfer (a.k.a domain adaptation) has been addressed for many tasks, including sentiment classification (Bollegala et al., 2011b), POS tagging (Schnabel and Schütze, 2013), syntactic parsing (Reichart and Rappoport, 2007; McClosky et al., 2010; Rush et al., 2012) and relation extraction (Jiang and Zhai, 2007; Bollegala et al., 2011a), if to name a handful of examples.

Several approaches to CD transfer have been proposed in the ML literature, including 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. Representation learning, the latter, has become prominent in the DNN era, and is the approach we take here. As noted in § 1 we adopt CD models that integrate pivot-based learning with DNNs to perform CLCD.

Multilingual word embeddings Multilingual word embeddings learning is an active field of research. For example, Ruder et al. (2017) compare 49 papers that have addressed the problem since 2011. Such embeddings are of importance as they provide means of bridging the lexical gap between languages, which supports CL transfer.

Surveying this extensive literature is well beyond our scope. Since our focus is on performing CLCD with minimal supervision, we quote Ruder et al. (2017) that categorize multilingual embedding methods with respect to two criteria on the data they require for their training: (a) type of alignment (word, sentence or document); and (b) comparability (parallel data: exact translation, vs. comparable data: data that is only similar). The BEs we use in our work are those of Smith et al. (2017) that require several thousands translated words as a supervision signal. That is, except from BEs induced using comparable word alignment signals – words aligned through indirect signals such as related images or through comparability of their features (e.g. POS tags) – the BEs we employ belong to the class of the most minimal supervision. In addition, as noted in § 1, in order to allow the lazy CLCD setup, we would like BEs where the source language embeddings are induced with no knowledge of the target language, and we indeed choose such BEs (§ 5).

3 Task Definition

The task we address is cross-language cross-domain (CLCD) learning. Formally, we are given a set of labeled examples from language L_s and domain D_s (denoted as the pair (L_s, D_s)). Our goal is to train an algorithm that will be able to correctly label examples from language L_t and domain D_t (L_t, D_t). The same label set, T, is used across the participating source and target domains and languages.

The setup we consider is similar in spirit to the setup known as unsupervised domain adaptation (e.g. (Blitzer et al., 2007; Ziser and Reichart, 2017, 2018)). When taking the representation learning approach to CLCD learning, the training pipeline usually consists of two steps. In the first step, the representation learning model is trained on unlabeled data from the source and target languages and domains, with the goal of generating a joint representation for the source and the target. Below we describe the unlabeled data in the full and the lazy CLCD setups. In the second step, a classifier for the supervised task is trained on the (L_s, D_s) labeled data. To facilitate language and domain transfer, every example that is fed to the task classifier in this second step is first represented by the representation model that was trained with unlabeled data at the first step. This is true both when the task classifier is trained and at test time when it is applied to data from (L_t, D_t).

We consider two setups which differ with respect to the unlabeled examples available for the representation learning model. In the full CLCD setup, the training algorithm has access to unlabeled examples from both (L_s, D_s) and (L_t, D_t). Since for truly resource poor languages it may be challenging to find a sufficient number of unlabeled examples from (L_t, D_t), we also consider the lazy setup where the training algorithm has access to unlabeled examples from (L_s, D_s) and (L_s, D_t) – that is, target domain unlabeled examples are available only in the source language.

4 Preliminaries

In this paper we aim to adapt CD models that integrate the power of DNNs and of pivot-based learning so that they can be applied to CLCD learning. In this section we hence briefly describe the works in this line. We start with the concept of domain adaptation using pivot-based methods, continue with works that are based on autoencoders and end with works that are based on sequence modeling with LSTMs.

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: (a) frequent in both domains; and (b) have a strong correlation with the task label in the source domain labeled data. The features which do not comply with at least one of these criteria form 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 cross-domain learning has been researched extensively (e.g. (Pan et al., 2010; Gouws et al., 2012; Bollegala et al., 2015; Yu and Jiang, 2016; Yang et al., 2017)).

4.1 Autoencoder Based Methods

An autoencoder (AE) is comprised of 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 model is trained to minimize a loss between x and $r(x)$. Over the last decade AEs have become prominent in CD learning with methods such as Stacked Denoising Autoencoders (SDA, (Vincent et al., 2008; Glorot et al., 2011) and marginalized SDA (MSDA, (Chen et al., 2012)) outperforming earlier state-of-the-art methods that were based on the concept of pivots but did not employ DNNs (Blitzer et al., 2006, 2007). A survey of AE-based models in CD learning can be found in ZR17.

ZR17 combined AEs and pivot-based modeling for CD learning. Their basic model (AE-SCL) is a feed-forward NN where the non-pivot features of the input example are encoded into a hidden representation that is then decoded into the pivot features of the example. Their advanced model (AE-SCL-SR) is identical in structure but its reconstruction matrix is fixed and consists of pre-trained embeddings of the pivot features, so that input examples with similar pivots are biased to have similar hidden representations. Since no CL learning was attempted in that work, the pre-trained embeddings used in AE-SCL-SR are monolingual. Both models are illustrated in Figure 1.

After one of the above representation models is trained with unlabeled data from the source and target domains, it is employed when training the task (sentiment analysis) classifier and when applying this classifier to test data. ZR17 learned a standard linear classifier (logistic regression), and fed it with the hidden representation of AE-SCL or AE-SCL-SR. They demonstrated the superiority of their models (especially, AE-SCL-SR) over non-DNN pivot-based methods and a variety of AE-based methods that do not consider pivots.

4.2 LSTM Based Methods

ZR18 observed that AE-based representation learning models do not exploit the structure of their input examples. Obviously, this can negatively impact text classification tasks, such as sentiment analysis. They hence proposed a structure-aware representation learning model, named Pivot Based Language Modeling (PBLM, Figure 2a).

PBLM is an LSTM fed with the embeddings of the input example words. As is standard in the LSTM literature, it is possible to feed the model with 1-hot word vectors and multiply them by a (randomly initialized) embeddings matrix (as done by ZR18) or to feed the model with pre-trained embeddings. In this paper we consider both options, taking advantage of the second in order to feed the model with BEs.

In contrast to standard LSTM-based language

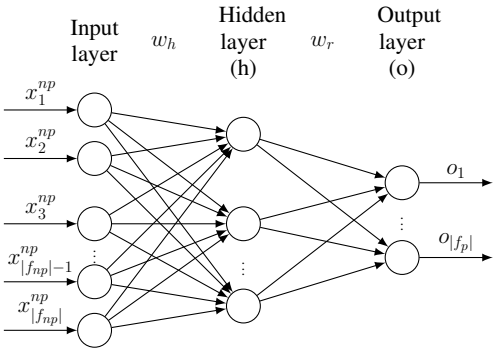

Figure 1: An illustrations of the AE-SCL and AE-SCL-SR models (figure imported from ZR17). x^{np} is a binary vector indicating whether each of the non-pivot features appears in the input example or not. x^p is a similar vector defined with respect to pivot features. o, the output vector of the model, provides the probability that each of the pivot features appears in the example, according to the model. The loss function of both models is the cross-entropy loss between o and x^p. While in AE-SCL both the encoding matrix w^h and the reconstruction matrix w^r are optimized, in AE-SCL-SR w^r consists of pre-trained word embeddings.

models that predict at each point the most likely next input word, PBLM predicts the next input unigram or bigram if one of these is a pivot (if both are, it predicts the bigram) and NONE otherwise. Similarly to AE-SCL and AE-SCL-SR, PBLM is trained with unlabeled data from both the source and target domains.

consider the example in Figure 2a, provided in ZR18 for adaptation of a sentiment classifier between English book reviews and English reviews of kitchen appliances. 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. Another example in ZR18 for the same domain pair (see Figure 1 in their paper) is: "I was at first very excited with my new Zyliss salad spinner - it is easy to spin and looks great", from this sentence 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 to facilitate adaptation between the domains.

PBLM can naturally feed a structure-aware task classifier. Particularly, in the PBLM-CNN architecture that we consider here (Figure 2b),[2] the PBLM's softmax layer (that computes the probabilities of each pivot to be the next unigram/bigram) is cut and a matrix whose columns are the PBLM's h_t vectors is fed to the CNN.

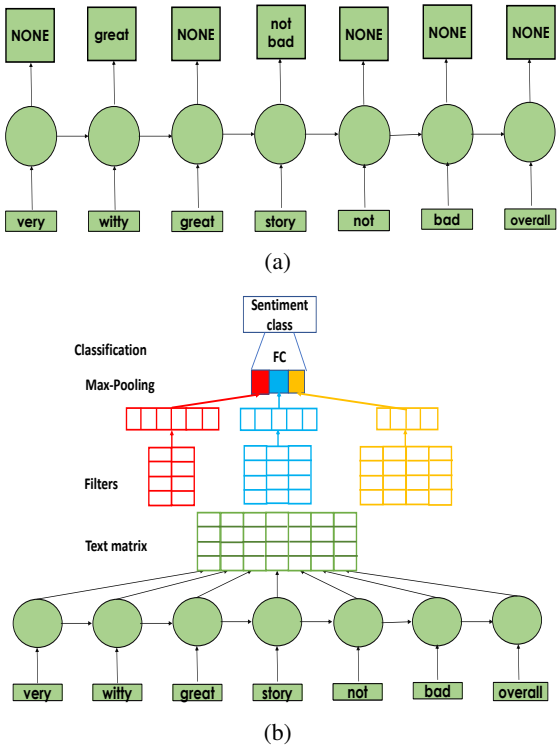

Figure 2: The PBLM model (figure imported form ZR18). (a) The PBLM representation learning model. (b) Adapting a classifier with PBLM: the PBLM-CNN model where PBLM representations are fed into a CNN task classifier.

ZR18 demonstrated the superiority of PBLM-CNN over AE-SCL, AE-SCL-SR and a variety of other previous models, emphasizing the importance of structure-awareness in CD transfer. We next discuss the adaptation of these models so that they can perform CLCD learning.

5 Cross-language Cross-domain Transfer

The models described in the previous section employ pivot-based learning (all models) and allow a convenient integration of BEs (AE-SCL-SR and PBLM). Below we discuss how we adapt these models so that they can perform CLCD learning.

[2] ZR18 also considered a PBLM-LSTM architecture where the PBLM representations feed an LSTM classifier. We focus on PBLM-CNN which demonstrated superior performance in 13 of 20 of their experimental setups.

Pivot translation We follow (Prettenhofer and Stein, 2010, 2011; Fernández et al., 2016) and redefine pivot features to be features that: (a) are frequent in (L_s, D_s) and that *their translation* is frequent in (L_t, D_t) ; and (b) are highly correlated with the task label in (L_s, D_s) . Note, that except for the translation requirement in (a) this is the classical definition of pivot features (§ 1).

Translated pivots are integrated into the models in a way that creates a shared cross-lingual output space. For both PBLM and the AE-based models a source language pivot feature and its translation are considered to be the same predicted class of the model. Consider, for example, a setup where we learn representations in order to adapt a classifier from (English, books) to (French, music). The pivot feature magnificent(English)/magnifique(French) will be considered the same PBLM prediction when trained on the unlabeled data from both (L_s, D_s) and (L_t, D_t). Similarly, in AE-SCL and AE-SCL-SR magnificent and magnifique will be assigned the same coordinate in the x^p (gold standard pivot indicators) and o (model output) vectors. In the lazy setup, where training is done with unlabeled data from (English, books) and (English, music) pivot translation is irrelevant as the representation learning model is trained only in the source language.

Note that when only pivot translation is used to make the CD methods address CLCD learning, the input space is not shared across languages. Instead, 1-hot vectors are used to encode the vocabularies of both languages, whose overlap is limited. This mismatch is somewhat reduced when training on unlabeled data from both (L_s, D_s) and (L_t, D_t). That is, we rely on the trained parameters of the models to align the input spaces when trained on unlabeled data from both (L_s, D_s) and (L_t, D_t).

In § 7 we show that this technique alone leads to improved CLCD results compared to existing methods. The lazy setup, however, is not supported by this technique, as training is not performed on unlabeled data from the target language. We next describe how to integrate BEs into our models, which provides a shared input layer that is crucial for both full and lazy CLCD.

Multilingual word embeddings Translated pivot features provide the models with a shared output layer. But can we use the same mechanism in order to map the input layers of the models into a shared cross-lingual space ?

Unfortunately, word-level translation does not seem like the right solution to this problem, due to two reasons. First, word-level translation is inherently ambiguous – it is very frequent that the set of senses associated with a given word in one language, is not identical to the set of senses associated with any other word in another given language. Moreover, large scale word-level translation may impose prohibitively high costs – either financial or in human time. Hence word-level translation is feasible mostly when dealing with a relatively small number of pivot features. The input layers of the models, consisting of words from the entire vocabulary (PBLM) or of non-pivot unigrams and bigrams (AE-SCL and AE-SCL-SR), require a cheaper and more stable mapping.

Our solution is hence based on BEs which embed words from the source and the target language in a shared vector space. As discussed in § 2 the BEs we use are those of Smith et al. (2017) that require several thousands of translated word pairs as a supervision signal, which reflects a low supervision level (Ruder et al., 2017). While bilingual word embedding models do not provide accurate word-level translation (to the level that such translation is possible), they do embed words from the two languages that have similar meaning with similar vectors, in terms of euclidean distance.

The BEs of Smith et al. (2017) also have the property required for our lazy setup: they are induced such that the source language embeddings have no knowledge of any particular target language. The embedding algorithm achieves that by learning two sets of monolingual embeddings and then aligning them with an SVD-based method.

Once we obtain the BEs, it is straightforward to integrate them into the PBLM model. We start by considering the full CLCD setup. When PBLM is applied to text from (L_s, D_s) – both when it is trained with unlabeled data (Figure 2a) and when it is used as part of the task classifier, when this classifier is trained with labeled data (Figure 2b) – the BEs of the source language words are fed into the model. Likewise, when PBLM is applied to text from (L_t, D_t) – both when it is trained with unlabeled data and when it is used as part of the task classifier when this classifier is applied to test data – it is fed with the bilingual representations of the target language words. In the lazy setup, the details are very similar except that PBLM is not

trained with unlabeled data from (L_t, D_t), only with unlabeled data from (L_s, D_s) and (L_s, D_t).

Unfortunately, BEs do not provide a sufficient solution for the AE-based models. In AE-SCL the input layer consists of a non-pivots indicator vector, x^{np}, that cannot be replaced by embedding vectors in a straight forward manner. In AE-SCL-SR the input layer is identical to that of AE-SCL, but this model replaces the reconstruction matrix w^r with a matrix whose rows consist of pre-trained word embeddings of the pivot features. Hence, similarly to PBLM we can construct a w^r matrix with the source language BEs when this model is applied to source language data, and with target language BEs when this model is applied to target language data. This construction of w^r provides an additional shared cross-lingual layer, added to the translated pivot features of the output layer.

Consequently, an inherent limitation of the AE-based models when it comes to CLCD transfer, is that they cannot be employed in the lazy setup. The intersection of their input spaces when applied to the source and the target languages is limited to the vectors representing the shared vocabulary items (see above in this section). Hence, these models have to be trained with unlabeled data from both languages in order to align the input layers of the two languages with each other.

6 Experiments

Task and data [3] As in our most related previous work (Prettenhofer and Stein, 2010, 2011; Fernández et al., 2016) we experiment with the Websis-CLS-10 dataset (Prettenhofer and Stein, 2010) consisting of Amazon product reviews written in 4 languages (English, German, French and Japanese), from 3 product domains (Books (B), DVDs (D) and Music (M)). Due to our extensive experimental setup we leave Japanese for future.[4]

For each (language, domain) pair the dataset includes 2000 train and 2000 test documents, labeled as positive or negative, and between 9,358 to 50,000 unlabeled documents. As in the aforementioned related works, we consider English as the source language, as it is likely to have labeled documents from the largest number of domains.

Following ZR18 we also consider a more challenging setup where the English source domain consists of user airline (A) reviews (Nguyen, 2015). We use the dataset of ZR18, consisting of 1000 positive and 1000 negative reviews in the labeled set, and 39396 reviews as the unlabeled set.

We employ a 5-fold cross-validation protocol. In all folds 1600 (English, D_s) train-set examples are randomly selected for training and 400 for development. The German and French test sets are used in all folds. All sets contain the same number of positive and negative reviews. For each model we report averaged performance across the folds.

The BEs were downloaded from the author's github. More details are in the appendix.

Models and baselines Our main model is PBLM+BE that is trained in the full setup and employs both translated pivots for CL output alignment and BEs for CL input alignment (§ 5). We also experiment with PBLM+BE+Lazy: the same model employed in the lazy setup, and with PBLM: a model similar to PBLM+BE except that BEs are not employed.

We further experiment with AE-SCL that employs translated pivots for CL output alignment and AE-SCL-SR that does the same and also integrates BEs into its fixed reconstruction matrix. Following ZR17 and ZR18 the linear classifier we use is logistic regression. To compare to previous work, we implemented the CL-SCL and the DCI models, for which we use the cosine kernel that performs best in (Fernández et al., 2016).

To consider the power of BEs, we experiment with a classifier fed with the BEs of the input document's words. We consider both a CNN classifier (where the BEs are fed into the columns of the CNN input matrix) and logistic regression (where the embeddings of the document's words are averaged) and report results with CNN as they are superior. We denote this model with BE+CNN.

For reference, we also compare to a setup where $L_s = L_t$, and to a setup where $L_s = L_t$ and $D_s = D_t$. For these setups we report results with a linear classifier with unigram and bigram features, as it outperforms both a linear classifier and a CNN with BE features. The models are denoted with Linear-IL and Linear-ILID, respectively (IL stands for in-language and ID for in-domain).

Pivot features For all models we consider unigrams and bigrams as features. To divide these

[3] The URLs of the code (previous models and standard packages) and data we used, are in the appendix.

[4] We add an English domain to our experiments. Moreover, training the models we consider here is substantially more time consuming as we employ DNNs, as opposed to previous methods that use linear classifiers.

Product Review Domains (Websis-CLS-10,(Prettenhofer and Stein, 2010)), CLCD														
	English-German							English-French						
Source-Target	**D-B**	**M-B**	**B-D**	**M-D**	**B-M**	**D-M**	**All**	**D-B**	**M-B**	**B-D**	**M-D**	**B-M**	**D-M**	**All**
PBLM Models														
PBLM+BE	**78.7**	**78.6**	**80.6**	**79.2**	**81.7**	**78.5**	**79.5**	**81.1**	**74.7**	**76.3**	**75.0**	**75.1**	**76.8**	**76.5**
PBLM	70.9	62.9	74.5	66.5	75.0	75.5	71.0	76.0	67.9	70.3	69.9	67.3	70.4	70.3
PBLM+BE+lazy	74.8	74.0	75.1	72.8	73.3	73.7	73.9	74.2	73.1	75.3	74.4	74.1	72.4	73.9
Autoencoder+pivot Models														
AE-SCL-SR	68.3	62.5	69.4	69.9	70.2	69	67.4	69.3	68.9	70.9	70.7	67	71.4	69.7
AE-SCL	67.9	63.7	68.7	63.8	69.0	70.1	67.2	68.6	66.1	69.2	69.4	66.7	68.1	68.0
Pivot-based (no DNN) Models														
CL-SCL	65.9	62.5	65.1	65.2	71.2	69.8	66.7	70.3	63.8	68.8	66.8	66.0	70.1	67.6
DCI	67.1	60.6	66.9	66.7	68.9	68.2	66.4	71.2	65.4	69.1	67.5	66.7	71.4	68.6
CLCD without CD Learning														
BE+CNN	62.8	63.8	65.3	68.7	71.6	72.0	67.3	69.5	59.7	63.7	65.7	65.9	67.0	65.2

Airline (English, (Nguyen, 2015)) to Product Review Domains (German or French), CLCD								
	English-German				English-French			
Source-Target	**A-B**	**A-D**	**A-M**	**All**	**A-B**	**A-D**	**A-M**	**All**
PBLM Models								
PBLM+BE	**67.9**	62.5	63.6	64.6	63.5	**66.9**	64.8	65.1
PBLM	60.9	59.6	60.1	60.2	60.9	61.9	58.9	60.5
PBLM+BE+lazy	66.3	**65.0**	**66.6**	**66.0**	**65.7**	65.6	**69.0**	**66.8**
Autoencoder+pivot Models								
AE-SCL-SR	55.8	57.5	60.8	58	55.8	52.9	56.3	55.7
AE-SCL	55.9	56.2	58.2	56.8	55.8	52.9	56.4	55.0
Pivot-based (no DNN) Models								
CL-SCL	56.6	52.6	53.7	54.3	52.7	54.5	53.1	53.4
DCI	55.9	52.1	54.5	54.1	53.1	53.7	53.9	53.5
CLCD without CD Learning								
BE+CNN	59.4	61.2	61.3	60.6	57.9	55.3	56.2	56.5

Product Review Domains (Websis-CLS-10,(Prettenhofer and Stein, 2010)), Within Language														
	German-German							French-French						
Source-Target	**D-B**	**M-B**	**B-D**	**M-D**	**B-M**	**D-M**	**All**	**D-B**	**M-B**	**B-D**	**M-D**	**B-M**	**D-M**	**All**
In-language cross-domain learning (no CD technique is employed)														
Linear-IL	81.5	78.9	77.8	76.7	77.6	79.8	78.7	80.2	78.2	79.2	79.7	78.5	79.7	79.3
In-language, In-domain learning														
Source-Target	**B-B**	–	**D-D**	–	**M-M**	–	**All**	**B-B**	–	**D-D**	–	**M-M**	–	**All**
Linear-ILID	84.2	–	81.5	–	83.3	–	83	84.1	–	79.2	–	85.8	–	83

Table 1: Sentiment accuracy. Top: CLCD transfer in the product domains. Middle: CLCD transfer from the English airline domain to the French and German product domains. Bottom: within language learning for the target languages. The "All" columns refer to the average over the setups in each table.

features into pivots and non-pivots we follow (Blitzer et al., 2007; Ziser and Reichart, 2017, 2018). Pivots are translated with Google translate. Pivot features are frequent in the unlabeled data of both the source and the target (language, domain) pairs: we require them to appear at least 10 times in each. Among those frequent features we select the ones with the highest mutual information with the task (sentiment) label in the source (language, domain) labeled data. For non-pivot features we consider unigrams and bigrams that appear at least 10 times in one of the (language, domain) pairs.[5]

[5]The average number of pivot features per review is ∼14 in the product to product experiments, and ∼11 when the airline domain and a product domain are involved.

Hyper-parameter tuning For all models we follow the tuning process described in the original papers. Details are in the appendix.

7 Results

Our results (Table 1) support the integration of structure-aware DNNs, translated pivots and BEs as advocated in this paper. Indeed, PBLM+BE which integrates all these factors and trained in the full setup is the best performing model in all 12 product setups (top table) and in 2 of 6 airline-product setups (middle table). PBLM+BE+lazy, the same model when trained in the lazy setup in which no target language unlabeled data is available for training, is the second best model in 9 of

12 product-product setups (in the other three setups only PBLM+BE and PBLM perform better) and is the best performing model in 4 of 6 airline-product setup and on average across these setups.

To better understand this last surprising result of the airline-product setups, we consider the pivot selection process (§ 6): (a) sort the source features by their mutual information with the source domain sentiment label; and (b) iterate over the pivots and exclude the ones whose translation frequency is not high enough in the target domain.

Let's examine the number of feature candidates that should be considered (in step (b)) from the list of criterion (a) in order to get 100 pivots. In product to product domain pairs: 182; In airline to product domain pairs: 304 (numbers are averaged across setups). In the lazy setup (where no pivot translation is performed) the corresponding numbers are: product to product domain pairs: 148; airline to product domain pairs: 173.

Hence, for domain pairs that involve airline and product, in the full setup many good pivots are *lost in translation* which affects the representation learning quality of PBLM+BE. While PBLM+BE+lazy does not get access to target language data, many more of its pivot features are preserved. We hypothesize that this can be one reason to the surprising superior performance of PBLM+BE+lazy when adapting from airline to product domains.

The success of PBLM+BE+lazy provides a particularly strong support to the validity of our approach, as this model lacks a major source of supervision available to the other CLCD models. As noted in § 1, we believe that the lazy setup is crucial for the future of CLCD learning.

Excluding BEs (PBLM) or changing the model to not generate a shared cross-lingual input layer (AE-SCL-SR that is also unaware of the review structure) results in substantial performance degradation. PBLM is better on average for all four CLCD setups, which emphasizes the importance of structure-awareness. Excluding both BEs and structure-awareness (AE) yields further degradation in most cases and on average. Yet, this degradation is minor (0.5% - 1.7% in the averages of the different setups), suggesting that the way AE-SCL-SR employs BEs, which is useful for CD transfer (ZR17), is less effective for CLCD.

CL-SCL and DCI, that employ pivot translation but neither DNNs nor BEs, lag behind the PBLM-based models and often also the AE-based models, although they outperform the latter in some cases. Likewise, BE+CNN, where BEs are employed but without any other CLCD learning technique, is also substantially outperformed by the PBLM-based models, but it does better than the AE-based models with the airline source domain.

Finally, comparison to the within-language models of the bottom table allows us to quantify the gap between current CLCD models and standard models that do not perform CD and/or CL transfer. The averaged differences between our best product-product model, PBLM-BE, and Linear-ILID are 3.5% (English-German) and 6.5% (English-French). When adapting from the airline domain the gap is much larger: averaged gaps of 17% and 16.2% from the best performing PBLM+BE-lazy, for English-German and English-French, respectively. This is not a surprise as ZR18 already demonstrated the challenging nature of within-language airline-product transfer. We consider our results to be encouraging, especially given the improvement over previous work, and the smaller gaps in the product-product setups.

8 Conclusions

We addressed the problem of CLCD transfer in sentiment analysis and proposed methods based on pivot-based learning, structure-aware DNNs and BEs. We considered full and lazy training, and designed a lazy model that, for a given target domain, can be trained with unlabeled data from the source language only and then be applied to any target language without re-training. Our models outperform previous models across 18 CLCD setups, even when ours are trained in the lazy setup and previous models are trained in the full setup.

In future work we wish to improve our results for large domain gaps and for more dissimilar languages, particularly in the important lazy setup. As our airline-product results indicate, increasing the domain gap harms our results, and we expect the same with more diverse language pairs.

Acknowledgements

We thank the anonymous reviewers and the members of the Technion NLP group for their useful comments. We also thank Ivan Vulić for his valuable guidance in the world of multilingual word embeddings.

References

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics* 4.

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, Fernando Pereira, et al. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. of ACL*.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proc. of EMNLP*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the ACL (TACL)* 5:135–146.

Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. 2015. Unsupervised cross-domain word representation learning. In *Proc. of ACL*.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2011a. Relation adaptation: learning to extract novel relations with minimum supervision. In *Proc. of IJCAI*.

Danushka Bollegala, David Weir, and John Carroll. 2011b. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *Proc. of ACL*.

Minmin Chen, Yixin Chen, and Kilian Q Weinberger. 2011. Automatic feature decomposition for single view co-training. In *Proc. of ICML*.

Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proc. of ICML*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proc. of ACL*.

Alejandro Moreo Fernández, Andrea Esuli, and Fabrizio Sebastiani. 2016. Distributional correspondence indexing for cross-lingual and cross-domain sentiment classification. *Journal of artificial intelligence research* 55(1):131–163.

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.

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*.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings ACL-IJCNLP)*.

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*.

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

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*.

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.

Peter Prettenhofer and Benno Stein. 2010. Cross-language text classification using structural correspondence learning. In *Proceedings of ACL*.

Peter Prettenhofer and Benno Stein. 2011. Cross-lingual adaptation using structural correspondence learning. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3(1):13.

Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proc. of ACL*.

Sebastian Ruder, Ivan Vuli, and Anders Sgaard. 2017. A survey of cross-lingual word embedding models. In *arXiv preprint arXiv:1706.04902*.

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*.

Tobias Schnabel and Hinrich Schütze. 2013. Towards robust cross-domain domain adaptation for part-of-speech tagging. In *Proc. of IJCNLP*.

Lei Shi, Rada Mihalcea, and Mingjun Tian. 2010. Cross language text classification by model translation and semi-supervised learning. In *Proceedings of EMNLP*.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *proceedings of ICLR*.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics* 1:1–12.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of ACL*.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proc. of ICML*.

Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In *Proceedings of ACL-IJCNLP*.

Wei Yang, Wei Lu, and Vincent Zheng. 2017. A simple regularization-based algorithm for learning cross-domain word embeddings. In *Proc. of EMNLP*.

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proc. of EMNLP*.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based lstm network for cross-lingual sentiment classification. In *Proceedings of EMNLP*.

Yftah Ziser and Roi Reichart. 2017. Neural structural correspondence learning for domain adaptation. In *Proc. of CoNLL*.

Yftah Ziser and Roi Reichart. 2018. Pivot based language modeling for improved neural domain adaptation. In *Proc. of NAACL-HLT*.

A Hyper-parameter Tuning

As promised in Section 6 of the main paper we detail here our hyper-parameter tuning process.

For all models, we tune the number of pivot features among $[100, 200, 300, 400, 500]$. For PBLM, the input embedding size (when no word embeddings are used) is tuned among $[128, 256]$, and the hidden representation dimension is selected from $[128, 256, 512]$. The size of the hidden layer of AE-SCL and AE-SCL-SR is set to 300.

The dimension of our bilingual embeddings is 300, as decided by (Smith et al., 2017). For all CNN models we use 256 filters of size $3 \times |embedding|$ and perform max pooling for each of the 256 vectors to generate a single 1×256 vector that is fed into the classification layer. In the SVD step of CL-SCL we tune the output dimension among $[50, 75, 100, 125, 150]$.

For AE-SCL and AE-SCL-SR, we follow ZR17 and represent each example fed into the sentiment classifier with its $w^h x^{np}$ vector. Unlike ZR17 we do not concatenate this representation with a bag of unigrams and bigrams representation of the example – due to the cross-lingual nature of our task. As in the original papers, the input features of AE-SCL, AE-SCL-SR, CL-SCL and DCI are word unigrams and bigrams.

All the algorithms in the paper that involve a CNN or a LSTM are trained with the ADAM algorithm (Kingma and Ba, 2015). For this algorithm we follow ZR18 and use 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: $decay = 0.0$.

B Code and Data

Here we provide the URLs of the code and data we used in this paper:

- The Websis-CLS-10 dataset (Prettenhofer and Stein, 2010) `http://www.uni-weimar.de/en/media/chairs/webis/research/corpora/corpus-webis-cls-10/`
- Bilingual word embeddings (Smith et al., 2017): `https://github.com/Babylonpartners/fastText_multilingual`. The authors employed their method to monolingual fastText embeddings (Bojanowski et al., 2017) – the embeddings of 78 languages were aligned with the English embeddings.
- The bilingual embeddings are based on the fastText Facebook embeddings (Bojanowski et al., 2017): `https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md`

- Logistic regression classifier: http://scikit-learn.org/stable/
- PBLM: We use the code from the author's github: https://github.com/yftah89/PBLM-Domain-Adaptation
- AE-SCL and AE-SCL-SR: We use the code from the author's github: https://github.com/yftah89/Neural-SCLDomain-Adaptation.
- We reimplemented the CL-SCL (Prettenhofer and Stein, 2011) and the DCI (Fernández et al., 2016) models.