TSP: Learning Task-Specific Pivots for Unsupervised Domain Adaptation

Xia Cui, Frans Coenen, and Danushka Bollegala

University of Liverpool

Abstract. Unsupervised Domain Adaptation (UDA) considers the problem of adapting a classifier trained using labelled training instances from a source domain to a different target domain, without having access to any labelled training instances from the target domain. Projection-based methods, where the source and target domain instances are first projected onto a common feature space on which a classifier can be trained and applied have produced state-of-the-art results for UDA. However, a critical pre-processing step required by these methods is the selection of a set of common features (aka. *pivots*), this is typically done using heuristic approaches,applied prior to performing domain adaptation. In contrast to the one of heuristics, we propose a method for learning Task-Specific Pivots (TSPs) in a systematic manner by considering both the labelled and unlabelled data available from both domains. We evaluate TSPs against pivots selected using alternatives in two cross-domain sentiment classification applications. Our experimental results show that the proposed TSPs significantly outperform previously proposed selection strategies in both tasks. Moreover, when applied in a cross-domain sentiment classification task, TSP captures many sentiment-bearing pivots.

1 Introduction

Domain Adaptation (DA) [5,7,30] considers the problem of adapting a model trained on one domain (*the source*) to a different domain (*the target*). DA is useful when we do not have sufficient labelled training data for a novel target domain to which we would like to apply a model that we have already trained using the labelled data for an existing source domain. If the source and target domains are sufficiently similar, then DA methods can produce accurate classifiers for the target domain [3]. DA methods have been widely applied for NLP tasks such as Part-Of-Speech (POS) tagging [32], sentiment classification [4], named entity recognition [22], and machine translation [24]. For example, in cross-domain sentiment classification, we might like to apply a sentiment classifier trained using labelled reviews on *books* for classifying reviews written on *Laptops*[1]

In Unsupervised Domain Adaptation (UDA), we assume the availability of unlabelled data from both source and target domains, and labelled data only from the source domain. In contrast, Supervised Domain Adaptation (SDA) assumes the availability of a small labelled dataset from the target domain [12,13]. UDA is a significantly harder task compared to SDA because of the unavailability of any labelled data from the target domain. In this paper, we focus on UDA. The main challenge of UDA is *feature mismatch* [6,8] – the features that appear in the source domain training instances are different from that in the target domain test instances. Because of the feature mismatch problem, even if we learn a highly accurate model from the source domain, most of those features will not affect in the target domain, resulting in a lower accuracy.

Current state-of-the-art methods for UDA first learn an embedding between the source and target domain feature spaces, and then learn a classifier for the target task (e.g. sentiment classification) in this embedded

[1] Here, a collection of reviews on a particular product category is considered as a *domain*.

	Books	*Laptops*
+	I think that this is an **excellent** book.	This is a **powerful**, yet **compact** laptop. An **excellent** choice for a travelling businessman!
-	This book is a **disappointment**, definitely **not recommended**.	It's the **worst** laptop I have ever had. It is **slow** and forever **crashing**.
-	Found myself skipping most of it found it **boring**.	**Pricy** laptop and does not deliver. A big **disappointment**. **Loud** fan, **noisy** hard drive. Never buy again.

Table 1: Positive (+) and negative (-) sentiment reviews on *Books* and *Laptops*. Sentiment-bearing features are shown in bold, whereas selected pivots are underlined.

space [5,6,30]. In order to learn this embedding, these methods must select a subset of common features (here onwards referred to as *pivots*) to the two domains. For example, consider the user reviews shown in Table 1 for *Books* and *Laptops* selected from Amazon[2]. Sentiment-bearing features, such as *powerful*, *loud*, *crashing*, *noisy* are specific to *laptops*, whereas *boring* would often be associated with a *book*. On the other hand, features such as *disappointment* and *excellent* are likely to be domain-independent, hence suitable as pivots for UDA.

As detailed later in Section 2, all existing strategies for selecting pivots are based on heuristics, such as selecting the top-frequent features that are common to both source and target domains [5]. In addition to frequency, Mutual Information (MI) [4], Pointwise Mutual Information (PMI) [6], and Positive Pointwise Mutual Information (PPMI) [9] have been proposed in prior work on UDA as pivot selection strategies.

There are two fundamental drawbacks associated with all existing heuristic-based pivot selection strategies. First, existing pivot selection strategies focus on either: (a) selecting a subset of the common features to source and target domains as pivots, or (b) selecting a subset of task-specific features (eg. sentiment-bearing features selected based on source domain's labelled training instances) as pivots. However, as we see later in our experiments, to successfully adapt to a new domain, the pivots must be both domain-independent (thereby capturing sufficient information for the knowledge transferring from the source to the target), as well as task-specific (thereby ensuring the selected pivots are accurately related to the labels). Second, it is non-trivial as to how we can combine the two requirements (a) and (b) in a consistent manner to select a set of pivots. Pivot selection can be seen as an optimal subset selection problem in which we must select a subset of the features from the intersection of the feature spaces of the two domains. Optimal subset selection is an NP-complete problem [14], and subset enumeration methods are practically infeasible considering that the number of subsets of a feature set of cardinality n is $2^n - 1$, where $n > 10^4$ in typical NLP tasks.

To overcome the above-mentioned limitations of existing pivot selection strategies, we propose **TSP** – Task-Specific Pivot selection for UDA. Specifically, we define two criteria for selecting pivots: one based on the *similarity between the source and target domains under a selected subset of features* (Section 3.2), and another based on *how well the selected subset of features capture the information related to the labelled instances in the source domain* (Section 3.3). We show that we can model the combination of the two criteria as a single constrained quadratic programming problem that can be efficiently solved for large feature spaces (Section 3.4). The reduction of pivot selection from a subset selection problem to a feature ranking problem, based on *pivothood* (α) scores learnt from the data, enables us to make this computation feasible. Moreover, the salience of each criteria can be adjusted via a *mixing parameter* (λ) enabling us to gradually increase the level of task-specificity in the selected pivots, which is not possible with existing heuristic-based pivot selection methods.

We compare the proposed TSP selection method against existing pivot selection strategies using two UDA methods: Spectral Feature Alignment (SFA) [30], and Structural Correspondence Learning (SCL) [4]

[2] `www.amazon.com`

on a benchmark dataset for cross-domain sentiment classification. We see that TSP, when initialised with pre-trained word embeddings trained using Continuous Bag-of-Words (CBOW) and Global Vector Prediction (GloVe) significantly outperforms previously proposed pivot selection strategies for UDA with respect to most domain pairs. Moreover, analysing the top-ranked pivots selected by TSP for their sentiment polarity, we see that more sentiment-bearing pivots are selected by TSP when we increase the mixing parameter. More importantly, our results show that we must consider both criteria discussed above when selecting pivots to obtain an optimal performance in UDA, which cannot be done using existing pivot selection strategies.

2 Related Work

In this section, we summarise previously proposed pivot selection strategies for UDA. For a detailed overview of DA methods, the intended reader is referred to [20].

Selecting common high-frequent features (FREQ) in source and target domains was proposed as a pivot selection strategy for cross-domain POS tagging by Blitzer et al. [5]. This strategy was used to select pivots for SCL to adapt a classification-based POS tagger. However, in their follow-up work Blitzer et al. [4] observed that for sentiment classification, where it is important to consider the polarity of the pivots, frequency is an inadequate criteria for pivot selection. To select pivots that behave similarly in the target domain as in the source domain, they computed the MI between a feature and source domain's positive vs. negative sentiment labelled reviews. If a particular feature was biased towards positive or negative labelled reviews, then it was likely that the feature contained information related to sentiment, which is useful for adapting a sentiment classifier for the target domain. Blitzer et al. [4] empirically showed MI to be a better pivot selection strategy than FREQ for cross-domain sentiment classification.

Pan et al. [30] proposed a modified version of MI where they weighted the MI between a feature and a domain by the probability of the feature in the domain to encourage domain-independent, as well as high-frequent, features to be selected as pivots. However, their pivot selection strategy used only the unlabelled data from the source and target domains. Unfortunately, there is no guarantee that the pivots selected purely based on unlabelled data will be related to the supervised target task (eg. sentiment classification or POS tagging) for which we plan to apply DA.

Bollegala et al. [6] proposed PMI [11] as a pivot selection strategy for UDA. Empirically, PMI gives a better normalisation result by considering individual feature occurrences. PPMI, which simply sets negative PMI values to zero has also been used as a pivot selection strategy for UDA [9]. Negative PMI values are often caused by noisy and unreliable co-occurrence counts; by ignoring these PPMI can select a more reliable set of pivots.

Overall, the above-mentioned heuristic-based pivot selection strategies can be seen as evaluating the dependence of a feature on: (a) source vs. target domain unlabelled training instances, or (b) source domain positive vs. negative labelled training instances. The above-mentioned four pivot selection strategies FREQ, MI, PMI and PPMI can be computed using either unlabelled data (corresponding to setting (a)) to derive four unlabelled versions of the pivot selection strategies FREQ_U, MI_U, PMI_U, and PPMI_U, or labelled data (corresponding to setting (b)) to derive four labelled versions of the pivot selection strategies FREQ_L, MI_L, PMI_L, and PPMI_L. In our experiments, we conducted an extensive comparison over all eight combinations, to the best of our knowledge, ours is the first paper to do so.

Although we focus on *feature-based* DA methods where the objective is to learn a common embedding between the source and target domain feature spaces, we note that there exist numerous *instance-based* DA methods that operate directly on training instances [21,22]. In instance-based DA, the goal is to select a subset of source domain labelled instances that is similar to the target domain unlabelled instance, and train a supervised classifier for the target task using only those selected instances. Sampling-based approaches [18] and weighting-based approaches have been proposed for instance-based DA [21].

Recently, deep learning approaches [10,16,17,27,33,34] have been proposed for further improving instance-based DA. Glorot et al. [17] used Stacked Denoising Auto-encoders (SDAs) to learn non-linear mappings for the data variations. Chen et al. [10] improved SDAs by marginalizing random corruptions under a specific network structure. Ganin et al. [16] proposed to regularise the immediate layers to perform feature learning, domain adaptation and classifier learning jointly using backpropagation for cross-domain image classification tasks.

Pivot selection does not apply for instance-based DA methods, hence not considered in the remainder of the paper.

3 Methods

3.1 Outline

Let us consider a source domain $\mathcal{S}$ consisting of a set of labelled instances $\mathcal{D}_L^{(S)}$ and unlabelled instances $\mathcal{D}_U^{(S)}$. Without loss of generality we assume the target adaptation task is binary classification, where we have a set of positively labelled instances $\mathcal{D}_+^{(S)}$ and a set of negatively labelled instances $\mathcal{D}_-^{(S)}$. Although we limit our discussion to the binary classification setting for simplicity, we note that the proposed method can be easily extended to other types of DA setting, such as multi-class classification and regression. For the target domain, denoted by $\mathcal{T}$, under UDA we assume the availability of only a set of unlabelled instances $\mathcal{D}_U^{(T)}$.

Given a pair of source and target domains, we propose a novel pivot selection method for UDA named **TSP** – Task Specific Pivot Selection. TSP considers two different criteria when selecting pivots.

First, we require that the *selected set of pivots must minimise the distance between the source and target domains*. The exact formulation of distance between domains and the corresponding optimisation problem are detailed in Section 3.2.

Second, we require that the *selected set of pivots be task-specific*. For example, if the target task is sentiment classification, then the selected set of pivots must contain sentiment bearing features. Given labelled data (annotated for the target task) from the source domain, we describe an objective function in Section 3.3 for this purpose.

Although the above-mentioned two objectives could be optimised separately, by jointly optimising for both objectives simultaneously we can obtain task-specific pivots that are also transferrable to the target domain. In Section 3.4, we formalise this joint optimisation problem and provide a quadratic programming-based solution. For simplicity, we present TSP for the pairwise UDA case, where we attempt to adapt from a single source domain to a single target domain. However, TSP can be easily extended to multi-source and multi-target UDA settings following the same optimisation procedure.

3.2 Pivots are Common to the Two Domains

Theoretical studies in UDA show that the upper bound on the classification accuracy on a target domain depends on the similarity between the source and the target domains [3]. Therefore, if we can somehow make the source and target domains similar by selecting a subset of the features from the intersection of their feature spaces, then we can learn better UDA models.

To formalise this idea into an objective we can optimise for, let us denote the possibility of a feature w_k getting selected as a pivot by $\alpha_k \in [0, 1]$. We refer to α_k as the *pivothood* of a feature $w_k \in \mathcal{W}$ ($|\mathcal{W}| = K$).

Higher pivothood values indicate that those features are more appropriate as pivots. The features could be, for example in sentiment classification, n-grams of words, POS tags, dependencies or any of their combinations. For simplicity of the disposition, let us assume that w_k are either unigrams or bigrams of words, and that we have pre-trained word embeddings $\boldsymbol{w}_k \in \mathbb{R}^D$ obtained using some word embedding learning algorithm. Our proposed method does not assume any specific properties of the word embeddings used; any fixed-dimensional representation of the features is sufficient, not limited to word embeddings.

Given source domain unlabelled data, we can compute, $\boldsymbol{c}^{(S)}$, the centroid for the source domain as follows:

$$\boldsymbol{c}^{(S)} = \frac{1}{\left|\mathcal{D}_U^{(S)}\right|} \sum_{d \in \mathcal{D}_U^{(S)}} \sum_{w_k \in d} \alpha_k \phi(w_k, d) \boldsymbol{w}_k \tag{1}$$

Here, $\phi(w_k, d)$ indicates the salience of w_k as a feature representing an instance d, such as the tf-idf measure popularly used in text classification. Likewise, we can compute the $\boldsymbol{c}^{(T)}$ centroid for the target domain using the unlabelled data from the target domain as follows:

$$\boldsymbol{c}^{(T)} = \frac{1}{\left|\mathcal{D}_U^{(T)}\right|} \sum_{d \in \mathcal{D}_U^{(T)}} \sum_{w_k \in d} \alpha_k \phi(w_k, d) \boldsymbol{w}_k \tag{2}$$

The centroid can be seen as a representation for the domain consisting all of the instances (reviews in the case of sentiment classification). If two domains are similar under some representation, then it will be easier to adapt from one domain to the other. Different distance measures can be used to compute the distance (or alternatively the similarity) between two domains under a particular representation such as ℓ_1 distance (Manhattan distance) or ℓ_2 distance (Euclidean distance). In this work, we use Euclidean distance for this purpose.

The problem of selecting a set of pivots can then be formulated as minimising the squared Euclidean distance between $\boldsymbol{c}^{(S)}$ and $\boldsymbol{c}^{(T)}$ given by,

$$\min\nolimits_{\boldsymbol{\alpha}} \left|\left|\boldsymbol{c}^{(S)} - \boldsymbol{c}^{(T)}\right|\right|_2^2 \tag{3}$$

Here, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)^\top$ is the pivothood indicator vector.

Assuming $\phi(w, d) = 0$ for $w \notin d$, and by substituting (1) and (2) in (3), we can re-write the objective as follows:

$$\min\nolimits_{\boldsymbol{\alpha}} \left|\left|\sum_{k=1}^{K} \alpha_k f(w_k) \boldsymbol{w}_k\right|\right|_2^2 \tag{4}$$

Here, $f(w_k)$ can be pre-computed and is given by,

$$f(w_k) = \frac{1}{\left|\mathcal{D}_U^{(S)}\right|} \sum_{d \in \mathcal{D}_U^{(S)}} \phi(w_k, d) - \frac{1}{\left|\mathcal{D}_U^{(T)}\right|} \sum_{d \in \mathcal{D}_U^{(T)}} \phi(w_k, d). \tag{5}$$

Minimisation of (4) can be trivially achieved by setting $\alpha_k = 0, \forall k = 1, \ldots, K$. To avoid this trivial solution and to make the objective scale invariant, we introduce the constraint $\sum_k \alpha_k = 1$.

Further, if we define $\boldsymbol{u}_k = f(w_k)\boldsymbol{w}_k$, then (4) reduces to the constrained least square regression problem given by (6).

$$\begin{aligned} &\min._{\boldsymbol{\alpha}} \left|\left|\sum_{k=1}^{K} \alpha_k \boldsymbol{u}_k\right|\right|_2^2 \\ &\text{s.t.} \sum_{k=1}^{K} \alpha_k = 1 \end{aligned} \tag{6}$$

3.3 Pivots are Task Specific

The objective defined in the previous Section is computed using only unlabelled data, hence agnostic to the target task. However, prior work on UDA [4,6] has shown that we must select pivots that are specific to the target task in order to be accurately adapted to cross-domain prediction tasks. Labelled data can be used to measure the task specificity of a feature. However, in UDA, the only labelled data we have is from the source domain. Therefore, we define the task specificity γ_k of a feature w_k as follows:

$$\gamma_k = \left(h(w_k, \mathcal{D}_+^{(S)}) - h(w_k, \mathcal{D}_-^{(S)})\right)^2 \tag{7}$$

Here, $h(w_k, \mathcal{D}_+^{(S)})$ denotes the association between the feature w_k and the source domain positive labelled instances. A wide range of association measures such as MI, PMI, PPMI, χ^2, log-likelihood ratio can be used as h [28]. PMI [11] between a feature w_k and a set of training instances $\mathcal{D}$ is given by:

$$\text{PMI}(w_k, \mathcal{D}) = \log\left(\frac{p(w_k, \mathcal{D})}{p(w_k)p(\mathcal{D})}\right), \tag{8}$$

We compute the probabilities p in (8) using frequency counts. PPMI [26] is a variation of PMI and follows PMI$(w_k, \mathcal{D})$ defined by (8), PPMI can be given by,

$$\text{PPMI}(w_k, \mathcal{D}) = \max(\text{PMI}(w_k, \mathcal{D}), 0) \tag{9}$$

We use PPMI as h in our experiments to reduce the noise caused by negative PMI values (Section 2).

Given the task specificity of features, we can formulate the problem of pivot selection as follows:

$$\min._{\boldsymbol{\alpha}} - \frac{\sum_k \alpha_k \gamma_k}{\sum_k \gamma_k} \tag{10}$$

3.4 Joint Optimisation

Ideally, we would like to select pivots that: (a) make source and target domain closer, as well as (b) are specific to the target task. A natural way to enforce both of those constraints is to linearly combine the two objectives given by (6) and (10) into the joint optimisation problem (11).

$$\begin{aligned} &\min._{\boldsymbol{\alpha}} \left|\left|\sum_{k=1}^{K} \alpha_k \boldsymbol{u}_k\right|\right|_2^2 - \lambda \frac{\sum_k \alpha_k \gamma_k}{\sum_k \gamma_k} \\ &\text{s.t.} \sum_{k=1}^{K} \alpha_k = 1 \end{aligned} \tag{11}$$

Here, the mixing parameter $\lambda \geq 0$ is a hyperparameter that controls the level of task specificity of the selected pivots. For example, by setting $\lambda = 0$ we can select pivots purely using unlabelled data. When we gradually increase λ, the source domain labelled data influences the pivot select process, making the selected pivots more task specific.

To further simplify the optimisation problem given by (11), let us define $\mathbf{U} \in \mathbb{R}^{D \times K}$ to be a matrix where the K columns correspond to $\boldsymbol{u}_k \in \mathbb{R}^D$, and $\boldsymbol{c}$ to be a vector whose k-th element is set to

$$c_k = -\lambda \gamma_k / \sum_k \gamma_k.$$

Then, (11) can be written as the following quadratic programming problem:

$$\begin{aligned} \min{}_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T \mathbf{U}^\top \mathbf{U} \boldsymbol{\alpha} + \mathrm{c}^T \boldsymbol{\alpha} \\ \text{s.t. } \boldsymbol{\alpha}^T \mathbf{1} = 1, \\ \boldsymbol{\alpha} \geqslant 0. \end{aligned} \tag{12}$$

We solve the quadratic programming problem given by (11) using the conjugate gradient (CG) method [29]. In practice, $D \ll K$, for which $\mathbf{U}^\top \mathbf{U} \in \mathbb{R}^{K \times K}$ is rank deficient and is not positive semidefinite. However, performing a small diagonalised Gaussian noise perturbation is sufficient in practice to obtain locally optimal solutions via CG that are sufficiently accurate for our datasets. We use CVXOPT[3] to solve (12) to obtain pivothoods α_k. We rank the features w_k in the descending order of their corresponding α_k and select the top-ranked features as pivots for an UDA method.

The computational complexity of the quadratic programming problem is dominated by the eigenvalue decomposition of $\mathbf{U}^\top \mathbf{U}$, which is of the dimensionality $K \times K$. Recall that K is the total number of features in the feature space representing the source and target domain instances. Computing the eigenvalue decomposition of a $K \times K$ square matrix is $\mathcal{O}(K^3)$. However, we can use randomised truncated Eigensolvers to compute the top-ranked eigenvalues (corresponding to the best pivots), without having to perform the full eigenvalue decomposition [19].

4 Experiments

Because the purpose of selecting pivots is to perform UDA, and because we cannot determine whether a particular feature is suitable as a pivot by manual observation, the most direct way to evaluate a pivot selection method is to use the pivots selected by that method in a state-of-the-art UDA method and evaluate the relative increase/decrease in performance in that DA task. For this purpose, we select SCL and SFA, which are UDA methods and perform a cross-domain sentiment classification task. The task here is to learn from labelled reviews from the source domain (a collection of reviews about a particular product) and predict sentiment for a different target domain (a collection of reviews about a different product). It is noteworthy that sentiment classification is used here purely as an evaluation task, and our goal is not to improve the performance of sentiment classification itself but to evaluate pivot selection methods. Therefore, we keep all other factors other than the pivots used by the UDA methods fixed during our evaluations.

We use the multi-domain sentiment dataset [4], which contains Amazon user reviews for the four product categories *Books* (**B**), *Electronic appliances* (**E**), *Kitchen appliances* (**K**), and *Dvds* (**D**). For each domain we have 1000 positive and 1000 negative reviews, and ca. 15,000 unlabelled reviews. We use the standard split of $800 \times 2 = 1600$ train and $200 \times 2 = 400$ test reviews. We generate 12 DA tasks by selecting one of the four

[3] `http://cvxopt.org/`

domains as the source and another as the target. We represent a review using a bag-of-features consisting of unigrams and bigrams, excluding a standard stop words list. We drop features that occur less than 5 times in the entire dataset to remove noise. A binary logistic regression classifier with an ℓ_2 regulariser is trained in each setting to develop a binary sentiment classifier. The regularisation coefficient is tuned using the *music* domain as a development domain, which is not part of the train/test data. Although different classifiers could be used in place of logistic regression, by using a simpler classifier we can readily evaluate the effect of the pivots on the UDA performance. The classification accuracy on the target domain's test labelled reviews is used as the evaluation measure.

On average, for a domain pair we have $K = 2648$ features. We set the number of pivots selected to the top-ranked 500 pivots in all domain pairs. Later in Section 4.4 we study the effect of the number of pivots on the performance of the proposed method. We use the publicly available $D = 300$ dimensional GloVe[4] (trained using 42B tokens from the Common Crawl) and CBOW[5] (trained using 100B tokens from Google News) embeddings as the word representations required by TSP. For bigrams not listed in the pretrained models, we sum the embeddings of constituent unigrams following prior work on compositional approaches for creating phrase (or sentence) embeddings using word embeddings [1,23]. We denote TSP trained using GloVe and CBOW embeddings respectively by **T-GloVe** and **T-CBOW**. By using two different types of word embeddings we can evaluate the dependance (if any) of TSP on a particular type of word embedding learning algorithm. Later in Section 4.4, we evaluate the effect of counting-based and prediction-based word embeddings on the performance of TSP when used for computing source and target centroids respectively in (1) and (2). We use the inverse document frequency (IDF) [31] as the feature salience ϕ and PPMI as h in our experiments.[6]

4.1 Cross-Domain Sentiment Classification

To evaluate the effect on performance of an UDA method when we select pivots using the proposed TSP and previously proposed pivots selection methods, we compare the performance of SCL and SFA in 12 different adaptation tasks from a source **S** to a target domain **T** as shown in Tables 2 and 3. We use the binomial exact test at two significance levels to evaluate statistical significance. The *no-adapt* (**NA**) lower-baselines, which simply applies a model trained using the source domain labelled data on the target domain's test data without performing DA, produced a near random-level performance indicating the importance of performing DA for obtaining good performance.

In SCL, T-CBOW reports the best performance in 8 domain-pairs, whereas T-GloVe in 3. Although in B-E and K-D pairs respectively PMI_U and PMI_L report the best results, the difference in performance with T-CBOW is not significant. Overall, T-CBOW is the best method in SCL closely followed by T-GloVe. For each of the previously proposed pivot selection methods except frequency, we see that the performance is always better when we use labelled data (denoted by subscript L) than unlabelled data (denoted by subscript U) to select pivots.

On the other hand in SFA, T-GloVe reports the best performance in 5 domain-pairs, whereas T-CBOW in 3. Among the previously proposed pivot selection methods, FREQ_U performs best in 5 domain-pairs. However, we see that overall, T-GloVe and T-CBOW outperform all the other pivot selection methods. Moreover, the difference between performance reported by T-CBOW and T-GloVe is not significant, indicating that TSP is relatively robust against the actual word embedding method being used.

[4] http://nlp.stanford.edu/projects/glove/

[5] https://code.google.com/archive/p/word2vec/

[6] Performance of TSP was found to be robust over a wide-range of ϕ and h combinations.

S-T	NA	SCL									
		FREQ$_L$	FREQ$_U$	MI$_L$	MI$_U$	PMI$_L$	PMI$_U$	PPMI$_L$	PPMI$_U$	T-CBOW	T-GloVe
B-E	52.03	69.75	68.25	68.75	65.75	69.50	**75.75***	69.50	67.50	73.25	75.25*
B-D	53.51	70.25	73.25	74.25	59.75	76.50	72.00	76.50	70.50	**77.50**	77.00
B-K	51.63	76.25	74.25	78.25	63.50	80.00*	79.50	80.00*	77.00	**83.25****	82.00**
E-B	51.02	60.50	65.25	66.25	55.75	64.75	63.00	64.25	60.50	**68.75**	67.25
E-D	50.94	68.00	67.75	68.00	66.25	70.50	67.00	71.50	65.50	73.25	**74.00***
E-K	56.00	81.00	80.50	82.50	80.50	86.25*	77.50	85.75*	77.25	**87.50****	**87.50****
D-B	52.50	72.00	69.25	72.00	56.25	74.75	68.50	75.75*	69.50	**76.75***	75.50*
D-E	53.25	71.75	70.50	74.25	66.00	74.25	65.25	74.00	65.25	**76.25**	75.75
D-K	54.39	70.75	75.25	74.00	57.25	80.50	77.25	80.25	79.75	83.00**	**84.00****
K-B	51.29	66.75	67.75	68.50	56.00	74.00*	70.00	74.00*	69.25	**75.25***	74.25*
K-E	54.86	74.00	74.25	75.50	78.00	80.00*	72.25	80.00*	71.75	**82.00****	81.25*
K-D	50.94	67.00	65.75	68.00	60.00	**71.50**	67.50	**71.50**	68.75	70.75	70.75
AVG	52.70	70.67	71.00	72.52	63.75	75.21	71.30	75.25	70.21	**77.30***	77.04*

Table 2: Classification accuracy of SCL using pivots selected by TSP (T-CBOW & T-GloVe) and prior methods. For each domain pair, the best results are given in bold font. The last row is the average across all the domain pairs. Statistically significant improvements over the FREQ_U baseline according to the binomial exact test, are shown by "*" and "**" respectively at $p = 0.01$ and $p = 0.001$ levels.

Overall, we see that SCL benefits more from accurate pivot selection than SFA. SCL learns separate linear predictors for each pivot using non-pivots (i.e. features other than pivots) as features, whereas SFA builds a bi-partite graph between pivots and non-pivots on which eigenvalue decomposition is applied to obtain a lower-dimensional embedding of pivots. Therefore, the effect of pivots on SCL is more direct and critical than in SFA.

4.2 Effect of the mixing parameter

To evaluate the effect of the mixing parameter on the pivots selected by TSP, we use the pivots selected by TSP for different λ values with SCL and SFA, and measure the classification accuracy on a target domain's sentiment labelled test reviews. Hyperparameters such as the number of singular vectors used in SCL and the number of latent dimensions used in SFA are tuned using the *music* development domain as the target for each of the four source domains. Due to the space limitations we show the results for adapting from the **D** to **K** pair with SCL and SFA in Figure 1. We see that when we do not require pivots to be task-specific (i.e. $\lambda = 0$) the accuracy is low. However, when we gradually increase λ thereby enforcing the criterion that the selected pivots must be also task-specific, we see a steady improvement in accuracy, which then saturates. SFA [30] constructs a bipartite graph based on the co-occurrence relationship between domain-independent (pivots) and domain-specific features (non-pivots) after pivot selection. FREQ$_U$ uses a larger dataset (S_U and T_U) compared to TSP (S_L and T_U), such that decreases the chance of rare co-occurrence may happen for a smaller dataset. For the same reason, adding γ that computes from an even smaller dataset (S_L only) does not help much in SFA.

This result shows the importance of considering both objectives (6) and (10) appropriately when selecting pivots, instead of focusing on only one of the two. Moreover, performance of TSP is relatively robust for $\lambda > 10^{-3}$, which is attractive because we do not have to fine-tune the mixing-parameter for each UDA setting.

S-T	NA	SFA									
		FREQ$_L$	FREQ$_U$	MI$_L$	MI$_U$	PMI$_L$	PMI$_U$	PPMI$_L$	PPMI$_U$	T-CBOW	T-GloVe
B-E	52.03	70.50	74.00	73.25	66.00	74.00	71.00	74.00	70.50	**74.75**	73.75
B-D	53.51	71.50	**78.00**	69.50	60.00	72.75	74.75	72.75	73.00	77.75	**78.00**
B-K	51.63	72.75	74.25	73.00	66.50	78.50	75.75	78.50	75.00	**81.00***	80.75*
E-B	51.02	64.75	64.50	64.00	57.25	65.75	59.00	64.00	57.75	64.50	**67.00**
E-D	50.94	67.50	**74.50**	63.25	60.75	71.50	65.00	71.50	61.50	73.50	71.50
E-K	56.00	81.00	82.50	78.25	71.75	85.50	79.00	85.25	78.00	**87.00**	85.50
D-B	52.50	74.25	**79.00**	69.50	62.00	73.50	73.00	73.75	67.75	**79.00**	78.00
D-E	53.25	72.50	75.50	71.75	65.75	69.00	68.75	69.00	66.50	74.25	**76.75**
D-K	54.39	73.75	76.75	74.75	56.50	81.00	79.75	81.00	79.00	81.50	**83.25***
K-B	51.29	67.75	70.00	69.00	58.00	66.50	**71.25**	66.50	**71.25**	69.00	70.00
K-E	54.86	80.50	**84.50**	79.25	70.25	77.25	71.75	77.25	72.50	80.50	79.75
K-D	50.94	67.25	**77.75**	67.75	60.50	68.00	71.00	68.00	71.00	72.25	72.50
AVG	52.70	72.00	75.94	71.10	62.94	73.60	71.67	73.46	70.31	76.25	**76.40**

Table 3: Classification accuracy of SFA using pivots selected by TSP (T-CBOW & T-GloVe) and prior methods. For each domain pair, the best results are given in bold font. The last row is the average across all the domain pairs. Statistically significant improvements over FREQ_U baseline according to the binomial exact test are shown by "*" and "**" respectively at $p = 0.01$ and $p = 0.001$ levels.

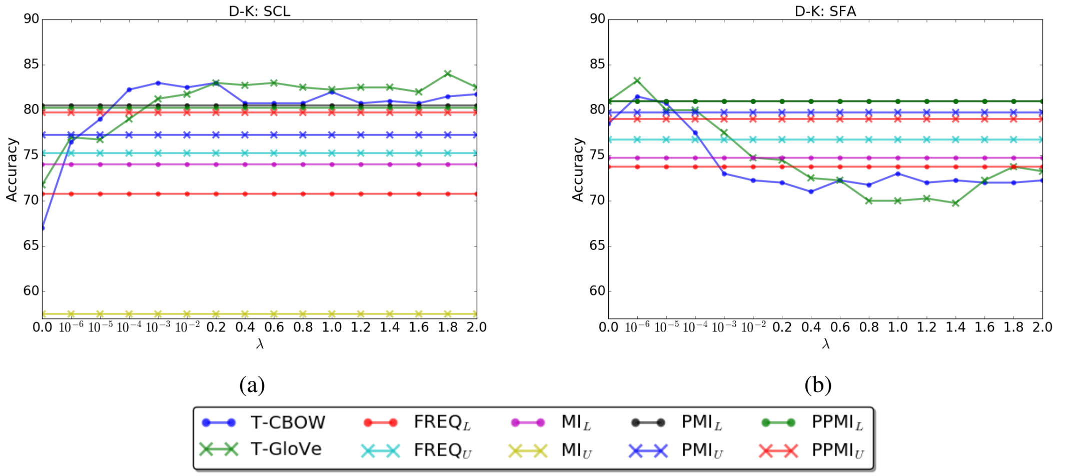
λ	T-CBOW	T-GloVe
0	unlike complaint fair i+havent name	wont whether yes complete so+good
10^{-5}	sent+it of+junk im+very fast+and an+excellent	go+wrong of+junk im+very i+called an+excellent
10^{-4}	of+junk value+for i+called fast+and it+comes	is+excellent sent+it im+very an+excellent good+price
10^{-3}	an+excellent of+junk i+called fast+and pleased+with	sent+it stopped+working of+junk is+perfect very+happy
10^{-2}	i+love of+junk dont+buy i+called very+happy	i+love of+junk i+called a+great very+happy
1.0	return stopped of+junk dont+buy i+called	stopped of+junk dont+buy i+called very+happy

Table 4: Top 5 pivots selected for adapting from **E** to **K** by T-CBOW and T-GloVe for different mixing parameter values. Bigrams are denoted by "+".

4.3 Sentiment Polarity of the Selected Pivots

To evaluate whether the pivots selected by TSP are task-specific, we conducted the following experiment. For a particular value of λ, we sort the features in the descending order of their α values, and select the top-ranked 500 features as pivots. Because our task in this case is sentiment classification, we would expect the selected pivots to be sentiment-bearing (i.e. containing words that express sentiment). Next, we compare the selected pivots against the sentiment polarity ratings provided in SentiWordNet [15]. SentiWordNet classifies each synset in the WordNet into positive, negative or neutral sentiment polarities such that any word with a positive sentiment will have a positive score, a negative sentiment a negative score, and zero otherwise. For bigrams not listed in the SentiWordNet, we compute the average polarity of the two constituent unigrams as the sentiment polarity of the bigrams. We consider a pivot to be task-specific if it has a non-zero polarity score.

Figure 2 shows the percentage of the task-specific (sentiment-bearing) pivots among the top-500 pivots selected by TSP for different λ values when adapting between **E** and **K**. We see that initially, when λ is small, the percentage of task-specific pivots is small. However, when we increase λ, thereby encouraging TSP to consider the task-specificity criterion more, we end up with pivots that are more sentiment-bearing.



(a)

(b)

Fig. 1: Accuracy of SCL and SFA when adapting from **D** source to **K** target (x-axis not to scale).

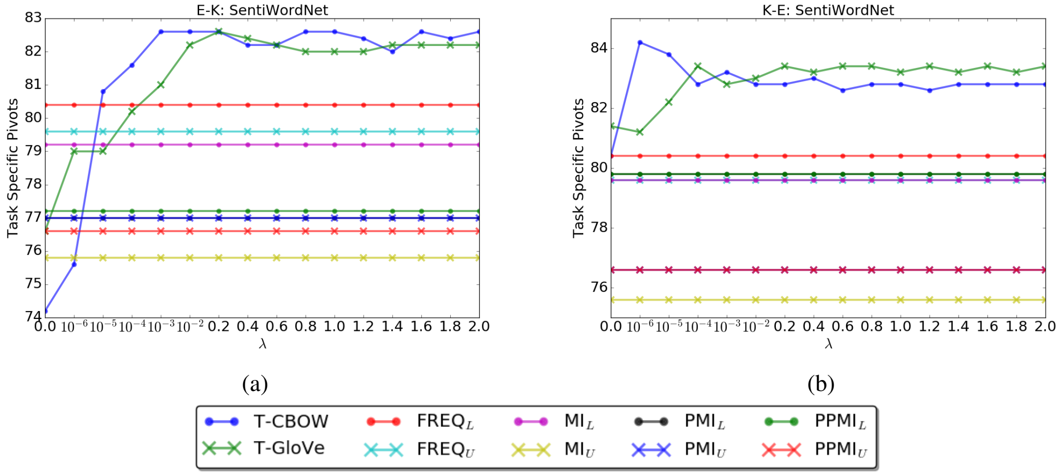
Moreover, when $\lambda > 10^{-3}$ the percentage of the task-specific pivots remains relatively stable, indicating that we have selected all pivots that are sentiment-bearing for the particular domain-pair.

As a qualitative example, we show the top-5 pivots ranked by their pivothood scores by T-CBOW and T-GloVe when adapting between **E** and **K** in Table 4. At $\lambda = 0$, we see that most of the top-ranked pivots are not sentiment-bearing. However, when we increase λ, we see that more and more sentiment-bearing pivots appear among the top-ranks. This result re-confirms that TSP can accurately capture task-specific pivots by jointly optimising for the two criteria proposed in Section 3. Interestingly, we see that many pivots are selected by both T-CBOW and T-GloVe such as *an+excellent*, *i+love*, and *very+happy*. This is encouraging because it shows that TSP depends weakly on the word representation method we use, thereby enabling us to potentially use a wide-range of existing pre-trained word embeddings with TSP.

4.4 Number of Selected Pivots and Effect of Word Embeddings

The number of pivots selected and the word embeddings are external inputs to TSP. In this section, we experimentally study the effect of the number of pivots selected by our proposed TSP and the word embeddings on the performance of the cross-domain sentiment classification. Specifically, we use TSP to select different k numbers of pivots, with three types of word embeddings: pre-trained word embeddings using CBOW (**T-CBOW**), pre-trained word embeddings using GloVe (**T-GloVe**), and counting-based word embeddings computed from Wikipedia (referred to as **T-Wiki**).

Counting-based word embeddings differ from prediction-based word embeddings such as CBOW and GloVe in several important ways [2,25]. In counting-based word embeddings we represent a target word by a vector where the elements correspond to words that co-occur with the target word in some contextual window. Entire sentences or windows of a fixed number of tokens can be considered as the co-occurrence window. Next, co-occurrences are aggregated over the entire corpus to build the final representation for the target word. Because any word can co-occur with the target word in some contextual window, counting-based word representations are often high dimensional (e.g. greater than 10^5), in comparison to prediction-based word embeddings (e.g. less than 1000 dimensions). Moreover, only a handful of words will co-occur with any



(a) (b)

Fig. 2: Task specific pivots for adapting between **E** and **K** by TSP with $\lambda \in [0, 2]$ (x-axis not to scale).

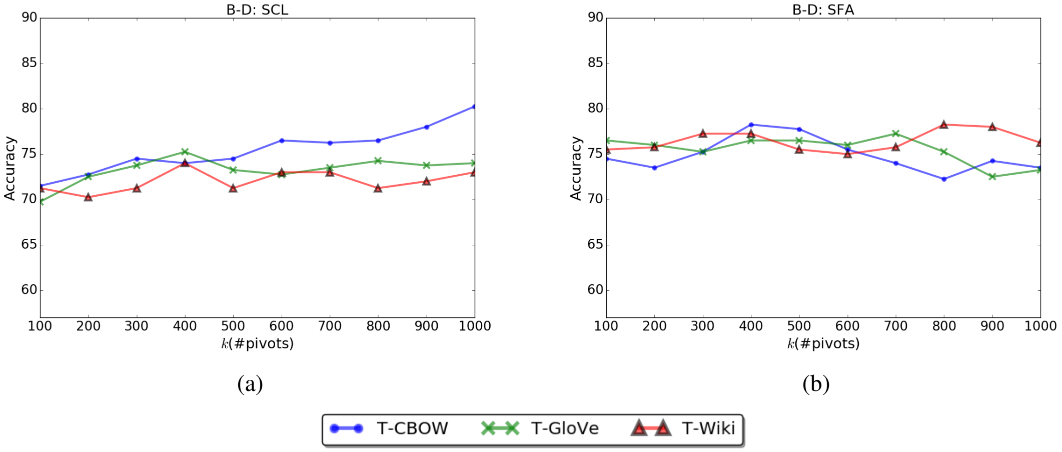
given target word even in a large corpus, producing highly sparse vectors. Unlike in prediction-based word embeddings where dimensions correspond to some latent attributes, counting-based word representations are easily interpretable because each dimension in the representation is explicitly assigned to a particular word in the vocabulary. By using both counting-based as well as prediction-based embeddings in the proposed method as the word embeddings used in (1) and (2), we can evaluate the effect of the word embeddings on the overall performance of the proposed method.

To build the counting-based embeddings (T-Wiki) we selected the January 2017 dump of English Wikipedia[7], and processed it using a Perl script[8] to create a corpus of 4.6 billion tokens. We select unigrams occurring at least 1000 times in this corpus amounting to a vocabulary of size $73,954$. We represent each word by a vector whose elements correspond to the PPMI values computed from the co-occurrence counts between words in this Wikipedia corpus.

To study the effect of the number of pivots k on the performance of TSP, we fix the mixing parameter $\lambda = 10^{-3}$ for all domain pairs and vary $k \in [100, 1000]$. We show the performance of SCL and SFA for the B-D pair respectively in Figures 3a and 3b. From Figure 3a, we see that, overall for SCL, T-CBOW outperforms the other two embeddings across a wide range of pivot set sizes. Moreover, its performance increases with k, which indicates that with larger pivot sets it can better represent a domain using the centroid. T-Wiki on the other hand reports lowest accuracies across all k values. Prior work evaluating word embedding learning algorithms has shown that prediction-based word embeddings outperform counting-based word embeddings for a wide-range of language processing tasks [2]. On the other hand, for SFA (Figure 3b) we do not see much difference in performance among the different word embeddings. Overall, the best performance is reported using **T-CBOW** with $k = 400$ pivots. This observation is in agreement with the recommendation by Pan et al. [30] to use 500 domain independent (pivot) features for this dataset. Similar trends were observed for all 12 domain pairs.

[7] `https://dumps.wikimedia.org`

[8] `http://mattmahoney.net/dc/textdata.html`



(a)

(b)

Fig. 3: Accuracy of SCL and SFA when adapting from **B** source to **D** target, from k (number of pivots) in the range $[100, 1000]$. The mixing parameter is set to $\lambda = 10^{-3}$.

5 Conclusion

We proposed TSP, a task-specific pivot selection method that simultaneously requires pivots to be both similar in the two domains as well as task-specific. TSP jointly optimises the two criteria by solving a single quadratic programming problem. The pivots selected by TSP improve the classification accuracy in multiple cross-domain sentiment classification tasks, consistently outperforming previously proposed pivot selection methods. Moreover, comparisons against SentiWordNet reveal that indeed the top-ranked pivots selected by TSP are task-specific. We conducted a series of experiments to study the behaviour of the proposed method with various parameters and the design choices involved such as the mixing parameter, number of pivots used, UDA method where the selected pivots are used, and the word embeddings used for representing features when computing the domain centroids. Our experimental results shows that TSP can find pivots for various pairs of domains and improve the performance of both SCL and SFA when compared to the performance obtained by using pivots selected by prior heuristics. Moreover, our analysis shows that it is important to use both unlabelled data (available for both source and target domains) as well as labelled data (available only for the source domain) when selecting pivots.

References

1. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: Proc. of ICLR (2017)
2. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 238–247. Association for Computational Linguistics, Baltimore, Maryland (June 2014)
3. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine Learning 79, 151–175 (2009)
4. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proc. of ACL. pp. 440–447 (2007)

5. Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. In: Proc. of EMNLP. pp. 120–128 (2006)
6. Bollegala, D., Mu, T., Goulermas, J.Y.: Cross-domain sentiment classification using sentiment sensitive embeddings. IEEE Transactions on Knowledge and Data Engineering 28(2), 398–410 (Feb 2015)
7. Bollegala, D., Weir, D., Carroll, J.: Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In: Proc. of ACL. pp. 132–141 (2011)
8. Bollegala, D., Weir, D., Carroll, J.: Cross-domain sentiment classification using a sentiment sensitive thesaurus. IEEE Transactions on Knowledge and Data Engineering 25(8), 1719 – 1731 (August 2013)
9. Bollegala, D., Weir, D., Carroll, J.: Learning to predict distributions of words across domains. In: Proc. of ACL. pp. 613 – 623 (2014)
10. Chen, M., Xu, Z., Weinberger, K., Sha, F.: Marginalized denoising autoencoders for domain adaptation. arXiv preprint arXiv:1206.4683 (2012)
11. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Computational Linguistics 16(1), 22 – 29 (March 1990)
12. Daumé III, H.: Frustratingly easy domain adaptation. In: Proc. of ACL. pp. 256–263 (2007)
13. Daumé III, H., Kumar, A., Saha, A.: Frustratingly easy semi-supervised domain adaptation. In: Proc. of the Workshop on Domain Adaptation for Natural Language Processing. pp. 53–59 (2010)
14. Davies, S., Russell, S.: Np-completeness of searches for smallest possible feature sets. In: Proc. of AAAI (1994)
15. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proc. of LREC. pp. 417–422 (2006)
16. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Proc. of ICML. pp. 1180 – 1189 (2015)
17. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: ICML'11 (2011)
18. Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In: Proc. of ICML (2013)
19. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: Probabilistic algorithms for constructung approximate matrix decompositions. SIAM REVIEW 53(2), 217 – 288 (2010)
20. Jiang, J.: A literature survey on domain adaptation of statistical classifiers. Tech. rep., UIUC (2008)
21. Jiang, J., Zhai, C.: Instance weighting for domain adaptation in nlp. In: Proc. of ACL. pp. 264 – 271 (2007)
22. Jiang, J., Zhai, C.: A two-stage approach to domain adaptation for statistical classifiers. In: Proc. of CIKM. pp. 401–410 (2007)
23. Kenter, T., Borisov, A., de Rijke, M.: Siamese cbow: Optimizing word embeddings for sentence representations. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 941–951. Association for Computational Linguistics, Berlin, Germany (August 2016)
24. Koehn, P., Schroeder, J.: Experiments in domain adaptation for statistical machine translation. In: Proc. of the Second Workshop on Statistical Machine Translation. pp. 224–227 (2007)
25. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Transactions of Association for Computational Linguistics 3, 211–225 (2015)
26. Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of the 17th international conference on Computational linguistics-Volume 2. pp. 768–774. Association for Computational Linguistics (1998)
27. Long, M., Wang, J.: Learning transferable features with deep adaptation networks. In: Proc. of ICML (2015)
28. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, Massachusetts (1999)
29. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer-Verlag (1999)
30. Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: Proc. of WWW. pp. 751–760 (2010)
31. Salton, G., Buckley, C.: Introduction to Modern Information Retreival. McGraw-Hill Book Company (1983)
32. Schnabel, T., Schütze, H.: Towards robust cross-domain domain adaptation for part-of-speech tagging. In: Proc. of IJCNLP. pp. 198–206 (2013)

33. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems. pp. 3320–3328 (2014)
34. Ziser, Y., Reichart, R.: Neural structural correspondence learning for domain adaptation. arXiv preprint arXiv:1610.01588 (2016)