Unsupervised Cross-Domain Word Representation Learning

Danushka Bollegala danushka.bollegala@liverpool.ac.uk University of Liverpool
Takanori Maehara maehara.takanori@shizuoka.ac.jp Shizuoka University
Ken-ichi Kawarabayashi k_keniti@nii.ac.jp National Institute of Informatics
JST, ERATO, Kawarabayashi Large Graph Project.

Abstract

Meaning of a word varies from one domain to another. Despite this important domain dependence in word semantics, existing word representation learning methods are bound to a single domain. Given a pair of *source-target* domains, we propose an unsupervised method for learning domain-specific word representations that accurately capture the domain-specific aspects of word semantics. First, we select a subset of frequent words that occur in both domains as *pivots*. Next, we optimize an objective function that enforces two constraints: (a) for both source and target domain documents, pivots that appear in a document must accurately predict the co-occurring non-pivots, and (b) word representations learnt for pivots must be similar in the two domains. Moreover, we propose a method to perform domain adaptation using the learnt word representations. Our proposed method significantly outperforms competitive baselines including the state-of-the-art domain-insensitive word representations, and reports best sentiment classification accuracies for all domain-pairs in a benchmark dataset.

1 Introduction

Learning semantic representations for words is a fundamental task in NLP that is required in numerous higher-level NLP applications (Collobert et al., 2011). Distributed word representations have gained much popularity lately because of their accuracy as semantic representations for words (Mikolov et al., 2013a; Pennington et al., 2014). However, the meaning of a word often varies from one domain to another. For example, the phrase *lightweight* is often used in a positive sentiment in the portable electronics domain because a lightweight device is easier to carry around, which is a positive attribute for a portable electronic device. However, the same phrase has a negative sentiment association in the *movie* domain because movies that do not invoke deep thoughts in viewers are considered to be lightweight (Bollegala et al., 2014). However, existing word representation learning methods are agnostic to such domain-specific semantic variations of words, and capture semantics of words only within a single domain. To overcome this problem and capture domain-specific semantic orientations of words, we propose a method that learns separate distributed representations for each domain in which a word occurs.

Despite the successful applications of distributed word representation learning methods (Pennington et al., 2014; Collobert et al., 2011; Mikolov et al., 2013a) most existing approaches are limited to learning only a single representation for a given word (Reisinger and Mooney, 2010). Although there have been some work on learning multiple *prototype* representations (Huang et al., 2012; Neelakantan et al., 2014) for a word considering its multiple senses, such methods do not consider the semantics of the domain in which the word is being used.

If we can learn separate representations for a word for each domain in which it occurs, we can use the learnt representations for domain adaptation tasks such as cross-domain sentiment classification (Bollegala et al., 2011b), cross-domain POS tagging (Schnabel and Schütze, 2013), cross-domain dependency parsing (McClosky et al., 2010), and domain adaptation of relation extractors (Bollegala et al., 2013a; Bollegala et al., 2013b; Bollegala et al., 2011a; Jiang and Zhai, 2007a; Jiang and Zhai, 2007b).

We introduce the *cross-domain word represen-*

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 730–740, Beijing, China, July 26-31, 2015.

tation learning task, where given two domains, (referred to as the *source* ($\mathcal{S}$) and the *target* ($\mathcal{T}$)) the goal is to learn two separate representations $\boldsymbol{w}_{\mathcal{S}}$ and $\boldsymbol{w}_{\mathcal{T}}$ for a word w respectively from the source and the target domain that capture *domain-specific* semantic variations of w. In this paper, we use the term *domain* to represent a collection of documents related to a particular topic such as user-reviews in Amazon for a product category (e.g. *books*, *dvds*, *movies*, etc.). However, a domain in general can be a field of study (e.g. *biology*, *computer science*, *law*, etc.) or even an entire source of information (e.g. *twitter*, *blogs*, *news articles*, etc.). In particular, we do not assume the availability of any labeled data for learning word representations.

This problem setting is closely related to unsupervised domain adaptation (Blitzer et al., 2006), which has found numerous useful applications such as, sentiment classification and POS tagging. For example, in unsupervised cross-domain sentiment classification (Blitzer et al., 2006; Blitzer et al., 2007), we train a binary sentiment classifier using positive and negative labeled user reviews in the source domain, and apply the trained classifier to predict sentiment of the target domain's user reviews. Although the distinction between the source and the target domains is not important for the word representation learning step, it is important for the domain adaptation tasks in which we subsequently evaluate the learnt word representations. Following prior work on domain adaptation (Blitzer et al., 2006), high-frequent features (unigrams/bigrams) common to both domains are referred to as *domain-independent* features or *pivots*. In contrast, we use *non-pivots* to refer to features that are specific to a single domain.

We propose an unsupervised cross-domain word representation learning method that jointly optimizes two criteria: (a) given a document d from the source or the target domain, we must accurately predict the non-pivots that occur in d using the pivots that occur in d, and (b) the source and target domain representations we learn for pivots must be similar. The main challenge in domain adaptation is *feature mismatch*, where the features that we use for training a classifier in the source domain do not necessarily occur in the target domain. Consequently, prior work on domain adaptation (Blitzer et al., 2006; Pan et al., 2010) learn lower-dimensional mappings from non-pivots to pivots, thereby overcoming the feature mismatch problem. Criteria (a) ensures that word representations for domain-specific non-pivots in each domain are related to the word representations for domain-independent pivots. This relationship enables us to discover pivots that are similar to target domain-specific non-pivots, thereby overcoming the feature mismatch problem.

On the other hand, criteria (b) captures the prior knowledge that high-frequent words common to two domains often represent domain-independent semantics. For example, in sentiment classification, words such as *excellent* or *terrible* would express similar sentiment about a product irrespective of the domain. However, if a pivot expresses different semantics in source and the target domains, then it will be surrounded by dissimilar sets of non-pivots, and reflected in the first criteria. Criteria (b) can also be seen as a regularization constraint imposed on word representations to prevent overfitting by reducing the number of free parameters in the model.

Our contributions in this paper can be summarized as follows.

- We propose a distributed word representation learning method that learns separate representations for a word for each domain in which it occurs. To the best of our knowledge, ours is the first-ever *domain-sensitive distributed* word representation learning method.

- Given domain-specific word representations, we propose a method to learn a cross-domain sentiment classifier.

  Although word representation learning methods have been used for various related tasks in NLP such as similarity measurement (Mikolov et al., 2013c), POS tagging (Collobert et al., 2011), dependency parsing (Socher et al., 2011a), machine translation (Zou et al., 2013), sentiment classification (Socher et al., 2011b), and semantic role labeling (Roth and Woodsend, 2014), to the best of our knowledge, word representations methods have not yet been used for cross-domain sentiment classification.

Experimental results for cross-domain sentiment classification on a benchmark dataset show that the word representations learnt using the proposed method statistically significantly outper-

form a state-of-the-art domain-insensitive word representation learning method (Pennington et al., 2014), and several competitive baselines. In particular, our proposed cross-domain word representation learning method is not specific to a particular task such as sentiment classification, and in principle, can be in applied to a wide-range of domain adaptation tasks. Despite this task-independent nature of the proposed method, it achieves the best sentiment classification accuracies on all domain-pairs, reporting statistically comparable results to the current state-of-the-art unsupervised cross-domain sentiment classification methods (Pan et al., 2010; Blitzer et al., 2006).

2 Related Work

Representing the semantics of a word using some algebraic structure such as a vector (more generally a tensor) is a common first step in many NLP tasks (Turney and Pantel, 2010). By applying algebraic operations on the word representations, we can perform numerous tasks in NLP, such as composing representations for larger textual units beyond individual words such as phrases (Mitchell and Lapata, 2008). Moreover, word representations are found to be useful for measuring semantic similarity, and for solving proportional analogies (Mikolov et al., 2013c). Two main approaches for computing word representations can be identified in prior work (Baroni et al., 2014): *counting-based* and *prediction-based*.

In counting-based approaches (Baroni and Lenci, 2010), a word w is represented by a vector $\boldsymbol{w}$ that contains other words that co-occur with w in a corpus. Numerous methods for selecting co-occurrence contexts such as proximity or dependency relations have been proposed (Turney and Pantel, 2010). Despite the numerous successful applications of co-occurrence counting-based distributional word representations, their high dimensionality and sparsity are often problematic in practice. Consequently, further post-processing steps such as dimensionality reduction, and feature selection are often required when using counting-based word representations.

On the other hand, prediction-based approaches first assign each word, for example, with a d-dimensional real-vector, and learn the elements of those vectors by applying them in an auxiliary task such as language modeling, where the goal is to predict the next word in a given sequence. The dimensionality d is fixed for all the words in the vocabulary, and, unlike counting-based word representations, is much smaller (e.g. $d \in [10, 1000]$ in practice) compared to the vocabulary size. The neural network language model (NNLM) (Bengio et al., 2003) uses a multi-layer feed-forward neural network to predict the next word in a sequence, and uses backpropagation to update the word vectors such that the prediction error is minimized.

Although NNLMs learn word representations as a by-product, the main focus on language modeling is to predict the next word in a sentence given the previous words, and not learning word representations that capture semantics. Moreover, training multi-layer neural networks using large text corpora is time consuming. To overcome those limitations, methods that specifically focus on learning word representations that model word co-occurrences in large corpora have been proposed (Mikolov et al., 2013a; Mnih and Kavukcuoglu, 2013; Huang et al., 2012; Pennington et al., 2014). Unlike the NNLM, these methods use *all* the words in a contextual window in the prediction task. Methods that use one or no hidden layers are proposed to improve the scalability of the learning algorithms. For example, the skip-gram model (Mikolov et al., 2013b) predicts the words c that appear in the local context of a word w, whereas the continuous bag-of-words model (CBOW) predicts a word w conditioned on all the words c that appear in w's local context (Mikolov et al., 2013a). Methods that use global co-occurrences in the entire corpus to learn word representations have shown to outperform methods that use only local co-occurrences (Huang et al., 2012; Pennington et al., 2014). Overall, prediction-based methods have shown to outperform counting-based methods (Baroni et al., 2014).

Despite their impressive performance, existing methods for word representation learning do not consider the semantic variation of words across different domains. However, as described in Section 1, the meaning of a word vary from one domain to another, and must be considered. To the best of our knowledge, the only prior work studying the problem of word representation variation across domains is due to Bollegala et al. (2014). Given a source and a target domain, they first select a set of pivots using pointwise mutual information, and create two distributional representa-

tions for each pivot using their co-occurrence contexts in a particular domain. Next, a projection matrix from the source to the target domain feature spaces is learnt using partial least squares regression. Finally, the learnt projection matrix is used to find the nearest neighbors in the source domain for each target domain-specific features. However, unlike our proposed method, their method *does not* learn domain-specific word representations, but simply uses co-occurrence counting when creating in-domain word representations.

Faralli et al. (2012) proposed a domain-driven word sense disambiguation (WSD) method where they construct glossaries for several domain using a pattern-based bootstrapping technique. This work demonstrates the importance of considering the domain specificity of word senses. However, the focus of their work is not to learn representations for words or their senses in a domain, but to construct glossaries. It would be an interesting future research direction to explore the possibility of using such domain-specific glossaries for learning domain-specific word representations.

Neelakantan et al. (2014) proposed a method that jointly performs WSD and word embedding learning, thereby learning multiple embeddings per word type. In particular, the number of senses per word type is automatically estimated. However, their method is limited to a single domain, and does not consider how the representations vary across domains. On the other hand, our proposed method learns a single representation for a particular word for each domain in which it occurs.

Although in this paper we focus on the monolingual setting where source and target domains belong to the same language, the related setting where learning representations for words that are translational pairs across languages has been studied (Hermann and Blunsom, 2014; Klementiev et al., 2012; Gouws et al., 2015). Such representations are particularly useful for cross-lingual information retrieval (Duc et al., 2010). It will be an interesting future research direction to extend our proposed method to learn such cross-lingual word representations.

3 Cross-Domain Representation Learning

We propose a method for learning word representations that are sensitive to the semantic variations of words across domains. We call this problem *cross-domain word representation learning*, and provide a definition in Section 3.1. Next, in Section 3.2, given a set of pivots that occurs in both a source and a target domain, we propose a method for learning cross-domain word representations. We defer the discussion of pivot selection methods to Section 3.4. In Section 3.5, we propose a method for using the learnt word representations to train a cross-domain sentiment classifier.

3.1 Problem Definition

Let us assume that we are given two sets of documents $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$ respectively for a source ($\mathcal{S}$) and a target ($\mathcal{T}$) domain. We do not consider the problem of retrieving documents for a domain, and assume such a collection of documents to be given. Then, given a particular word w, we define cross-domain representation learning as the task of learning two separate representations $\boldsymbol{w}_\mathcal{S}$ and $\boldsymbol{w}_\mathcal{T}$ capturing w's semantics in respectively the source $\mathcal{S}$ and the target $\mathcal{T}$ domains.

Unlike in domain adaptation, where there is a clear distinction between the source (i.e. the domain on which we train) vs. the target (i.e. the domain on which we test) domains, for representation learning purposes we do not make a distinction between the two domains. In the *unsupervised* setting of the cross-domain representation learning that we study in this paper, we do not assume the availability of labeled data for any domain for the purpose of learning word representations. As an extrinsic evaluation task, we apply the trained word representations for classifying sentiment related to user-reviews (Section 3.5). However, for this evaluation task we require sentiment-labeled user-reviews from the source domain.

Decoupling of the word representation learning from any tasks in which those representations are subsequently used, simplifies the problem as well as enables us to learn *task-independent* word representations with potential generic applicability. Although we limit the discussion to a pair of domains for simplicity, the proposed method can be easily extended to jointly learn word representations for more than two domains. In fact, prior work on cross-domain sentiment analysis show that incorporating multiple source domains improves sentiment classification accuracy on a target domain (Bollegala et al., 2011b; Glorot et al., 2011).

3.2 Proposed Method

To describe our proposed method, let us denote a pivot and a non-pivot feature respectively by c and w. Our proposed method does not depend on a specific pivot selection method, and can be used with all previously proposed methods for selecting pivots as explained later in Section 3.4. A pivot c is represented in the source and target domains respectively by vectors $\boldsymbol{c}_{\mathcal{S}} \in \mathbb{R}^n$ and $\boldsymbol{c}_{\mathcal{T}} \in \mathbb{R}^n$. Likewise, a source specific non-pivot w is represented by $\boldsymbol{w}_{\mathcal{S}}$ in the source domain, whereas a target specific non-pivot w is represented by $\boldsymbol{w}_{\mathcal{T}}$ in the target domain. By definition, a non-pivot occurs only in a single domain. For notational convenience we use w to denote non-pivots in both domains when the domain is clear from the context. We use $\mathcal{C}_{\mathcal{S}}$, $\mathcal{W}_{\mathcal{S}}, \mathcal{C}_{\mathcal{T}}$, and $\mathcal{W}_{\mathcal{T}}$ to denote the sets of word representation vectors respectively for the source pivots, source non-pivots, target pivots, and target non-pivots.

Let us denote the set of documents in the source and the target domains respectively by $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{T}}$. Following the bag-of-features model, we assume that a document D is represented by the set of pivots and non-pivots that occur in D ($w \in d$ and $c \in d$). We consider the co-occurrences of a pivot c and a non-pivot w within a fixed-size contextual window in a document. Following prior work on representation learning (Mikolov et al., 2013a), in our experiments, we set the window size to 10 tokens, without crossing sentence boundaries. The notation $(c, w) \in d$ denotes the co-occurrence of a pivot c and a non-pivot w in a document d.

We learn domain-specific word representations by maximizing the prediction accuracy of the non-pivots w that occur in the local context of a pivot c. The hinge loss, $L(\mathcal{C}_{\mathcal{S}}, \mathcal{W}_{\mathcal{S}})$, associated with predicting a non-pivot w in a source document $d \in \mathcal{D}_{\mathcal{S}}$ that co-occurs with pivots c is given by:

$$\sum_{d \in \mathcal{D}_{\mathcal{S}}} \sum_{(c,w) \in d} \sum_{w^* \sim p(w)} \max\left(0, 1 - \boldsymbol{c}_{\mathcal{S}}{}^{\top} \boldsymbol{w}_{\mathcal{S}} + \boldsymbol{c}_{\mathcal{S}}{}^{\top} \boldsymbol{w}^*_{\mathcal{S}}\right) \tag{1}$$

Here, $\boldsymbol{w}^*_{\mathcal{S}}$ is the source domain representation of a non-pivot w^* that *does not occur* in d. The loss function given by Eq. 1 requires that a non-pivot w that co-occurs with a pivot c in the document d is assigned a higher ranking score as measured by the inner-product between $\boldsymbol{c}_{\mathcal{S}}$ and $\boldsymbol{w}_{\mathcal{S}}$ than a non-pivot w^* that does not occur in d. We randomly sample k non-pivots from the set of all source domain non-pivots that do not occur in d as w^*.

Specifically, we use the marginal distribution of non-pivots $p(w)$, estimated from the corpus counts, as the sampling distribution. We raise $p(w)$ to the $3/4$-th power as proposed by Mikolov et al. (2013a), and normalize it to unit probability mass prior to sampling k non-pivots w^* per each co-occurrence of $(c, w) \in d$. Because non-occurring non-pivots w^* are randomly sampled, prior work on noise contrastive estimation has found that it requires more negative samples than positive samples to accurately learn a prediction model (Mnih and Kavukcuoglu, 2013). We experimentally found $k = 5$ to be an acceptable trade-off between the prediction accuracy and the number of training instances.

Likewise, the loss function $L(\mathcal{C}_{\mathcal{T}}, \mathcal{W}_{\mathcal{T}})$ for predicting non-pivots using pivots in the target domain is given by:

$$\sum_{d \in \mathcal{D}_{\mathcal{T}}} \sum_{(c,w) \in d} \sum_{w^* \sim p(w)} \max\left(0, 1 - \boldsymbol{c}_{\mathcal{T}}{}^{\top} \boldsymbol{w}_{\mathcal{T}} + \boldsymbol{c}_{\mathcal{T}}{}^{\top} \boldsymbol{w}^*_{\mathcal{T}}\right) \tag{2}$$

Here, w^* denotes target domain non-pivots that *do not occur* in d, and are randomly sampled from $p(w)$ following the same procedure as in the source domain.

The source and target loss functions given respectively by Eqs. 1 and 2 can be used on their own to independently learn source and target domain word representations. However, by definition, pivots are common to both domains. We use this property to relate the source and target word representations via a *pivot-regularizer*, $R(\mathcal{C}_{\mathcal{S}}, \mathcal{C}_{\mathcal{T}})$, defined as:

$$R(\mathcal{C}_{\mathcal{S}}, \mathcal{C}_{\mathcal{T}}) = \frac{1}{2} \sum_{i=1}^{K} ||\boldsymbol{c}^{(i)}_{\mathcal{S}} - \boldsymbol{c}^{(i)}_{\mathcal{T}}||^2 \tag{3}$$

Here, $||\boldsymbol{x}||$ represents the l_2 norm of a vector $\boldsymbol{x}$, and $c^{(i)}$ is the i-th pivot in a total collection of K pivots. Word representations for non-pivots in the source and target domains are linked via the pivot regularizer because, the non-pivots in each domain are predicted using the word representations for the pivots in each domain, which in turn are regularized by Eq. 3. The overall objective function, $L(\mathcal{C}_{\mathcal{S}}, \mathcal{W}_{\mathcal{S}}, \mathcal{C}_{\mathcal{T}}, \mathcal{W}_{\mathcal{T}})$, we minimize is the sum[1] of

[1] Weighting the source and target loss functions by the respective dataset sizes did not result in any significant increase in performance. We believe that this is because the benchmark dataset contains approximately equal numbers of documents for each domain.

the source and target loss functions, regularized via Eq. 3 with coefficient λ, and is given by:

$$L(\mathcal{C}_{\mathcal{S}}, \mathcal{W}_{\mathcal{S}},) + L(\mathcal{C}_{\mathcal{T}}, \mathcal{W}_{\mathcal{T}}) + \lambda R(\mathcal{C}_{\mathcal{S}}, \mathcal{C}_{\mathcal{T}}) \quad (4)$$

3.3 Training

Word representations of pivots c and non-pivots w in the source ($\boldsymbol{c}_{\mathcal{S}}$, $\boldsymbol{w}_{\mathcal{S}}$) and the target ($\boldsymbol{c}_{\mathcal{T}}$, $\boldsymbol{w}_{\mathcal{T}}$) domains are parameters to be learnt in the proposed method. To derive parameter updates, we compute the gradients of the overall loss function in Eq. 4 w.r.t. to each parameter as follows:

$$\frac{\partial L}{\partial \boldsymbol{w}_{\mathcal{S}}} = \begin{cases} 0 & \text{if } \boldsymbol{c}_{\mathcal{S}}{}^{\top}(\boldsymbol{w}_{\mathcal{S}} - \boldsymbol{w}^*_{\mathcal{S}}) \geq 1 \\ -\boldsymbol{c}_{\mathcal{S}} & \text{otherwise} \end{cases} \quad (5)$$

$$\frac{\partial L}{\partial \boldsymbol{w}^*_{\mathcal{S}}} = \begin{cases} 0 & \text{if } \boldsymbol{c}_{\mathcal{S}}{}^{\top}(\boldsymbol{w}_{\mathcal{S}} - \boldsymbol{w}^*_{\mathcal{S}}) \geq 1 \\ \boldsymbol{c}_{\mathcal{S}} & \text{otheriwse} \end{cases} \quad (6)$$

$$\frac{\partial L}{\partial \boldsymbol{w}_{\mathcal{T}}} = \begin{cases} 0 & \text{if } \boldsymbol{c}_{\mathcal{T}}{}^{\top}(\boldsymbol{w}_{\mathcal{T}} - \boldsymbol{w}^*_{\mathcal{T}}) \geq 1 \\ -\boldsymbol{c}_{\mathcal{T}} & \text{otherwise} \end{cases} \quad (7)$$

$$\frac{\partial L}{\partial \boldsymbol{w}^*_{\mathcal{T}}} = \begin{cases} 0 & \text{if } \boldsymbol{c}_{\mathcal{T}}{}^{\top}(\boldsymbol{w}_{\mathcal{T}} - \boldsymbol{w}^*_{\mathcal{T}}) \geq 1 \\ \boldsymbol{c}_{\mathcal{T}} & \text{otherwise} \end{cases} \quad (8)$$

$$\frac{\partial L}{\partial \boldsymbol{c}_{\mathcal{S}}} = \begin{cases} \lambda(\boldsymbol{c}_{\mathcal{S}} - \boldsymbol{c}_{\mathcal{T}}) & \text{if } \boldsymbol{c}_{\mathcal{S}}{}^{\top}(\boldsymbol{w}_{\mathcal{S}} - \boldsymbol{w}^*_{\mathcal{S}}) \geq 1 \\ \boldsymbol{w}^*_{\mathcal{S}} - \boldsymbol{w}_{\mathcal{S}} + \lambda(\boldsymbol{c}_{\mathcal{S}} - \boldsymbol{c}_{\mathcal{T}}) & \text{otherwise} \end{cases} \quad (9)$$

$$\frac{\partial L}{\partial \boldsymbol{c}_{\mathcal{T}}} = \begin{cases} \lambda(\boldsymbol{c}_{\mathcal{T}} - \boldsymbol{c}_{\mathcal{S}}) & \text{if } \boldsymbol{c}_{\mathcal{T}}{}^{\top}(\boldsymbol{w}_{\mathcal{T}} - \boldsymbol{w}^*_{\mathcal{T}}) \geq 1 \\ \boldsymbol{w}^*_{\mathcal{T}} - \boldsymbol{w}_{\mathcal{T}} + \lambda(\boldsymbol{c}_{\mathcal{T}} - \boldsymbol{c}_{\mathcal{S}}) & \text{otherwise} \end{cases} \quad (10)$$

Here, for simplicity, we drop the arguments inside the loss function and write it as L. We use mini batch stochastic gradient descent with a batch size of 50 instances. AdaGrad (Duchi et al., 2011) is used to schedule the learning rate. All word representations are initialized with n dimensional random vectors sampled from a zero mean and unit variance Gaussian. Although the objective in Eq. 4 is not jointly convex in all four representations, it is convex w.r.t. the representation of a particular feature (pivot or non-pivot) when the representations for all the other features are held fixed. In our experiments, the training converged in all cases with less than 100 epochs over the dataset.

The rank-based predictive hinge loss (Eq. 1) is inspired by the prior work on word representation learning for a single domain (Collobert et al., 2011). However, unlike the multilayer neural network in Collobert et al. (2011), the proposed method uses a computationally efficient single layer to reduce the number of parameters that must be learnt, thereby scaling to large datasets. Similar to the skip-gram model (Mikolov et al., 2013a), the proposed method predicts occurrences of contexts (non-pivots) w within a fixed-size contextual window of a target word (pivot) c.

Scoring the co-occurrences of two words c and w by the bilinear form given by the inner-product is similar to prior work on domain-insensitive word-representation learning (Mnih and Hinton, 2008; Mikolov et al., 2013a). However, unlike those methods that use the softmax function to convert inner-products to probabilities, we directly use the inner-products without any further transformations, thereby avoiding computationally expensive distribution normalizations over the entire vocabulary.

3.4 Pivot Selection

Given two sets of documents $\mathcal{D}_{\mathcal{S}}$, $\mathcal{D}_{\mathcal{T}}$ respectively for the source and the target domains, we use the following procedure to select pivots and non-pivots. First, we tokenize and lemmatize each document using the Stanford CoreNLP toolkit[2]. Next, we extract unigrams and bigrams as features for representing a document. We remove features listed as stop words using a standard stop words list. Stop word removal increases the effective co-occurrence window size for a pivot. Finally, we remove features that occur less than 50 times in the entire set of documents.

Several methods have been proposed in the prior work on domain adaptation for selecting a set of pivots from a given pair of domains such as the minimum frequency of occurrence of a feature in the two domains, mutual information (MI), and the entropy of the feature distribution over the documents (Pan et al., 2010). In our preliminary experiments, we discovered that a normalized version of the PMI (NPMI) (Bouma, 2009) to work consistently well for selecting pivots from different pairs of domains. NPMI between two features x and y is given by:

$$\text{NPMI}(x, y) = \log\left(\frac{p(x, y)}{p(x)p(y)}\right) \frac{1}{-\log(p(x, y))} \quad (11)$$

Here, the joint probability $p(x, y)$, and the marginal probabilities $p(x)$ and $p(y)$ are estimated using the number of co-occurrences of x and y in the sentences in the documents. Eq. 11 normalizes both the upper and lower bounds of the PMI.

[2] http://nlp.stanford.edu/software/corenlp.shtml

We measure the appropriateness of a feature as a pivot according to the score given by:

$$\text{score}(x) = \min\left(\text{NPMI}(x, \mathcal{S}), \text{NPMI}(x, \mathcal{T})\right). \quad (12)$$

We rank features that are common to both domains in the descending order of their scores as given by Eq. 12, and select the top $N_{\mathcal{P}}$ features as pivots. We rank features x that occur only in the source domain by $\text{NPMI}(x, \mathcal{S})$, and select the top ranked $N_{\mathcal{S}}$ features as source-specific non-pivots. Likewise, we rank the features x that occur only in the target domain by $\text{NPMI}(x, \mathcal{T})$, and select the top ranked $N_{\mathcal{T}}$ features as target-specific non-pivots.

The pivot selection criterion described here differs from that of Blitzer et al. (2006; 2007), where pivots are defined as features that behave similarly both in the source and the target domains. They compute the mutual information between a feature (i.e. unigrams or bigrams) and the sentiment labels using source domain labeled reviews. This method is useful when selecting pivots that are closely associated with positive or negative sentiment in the source domain. However, in unsupervised domain adaptation we do not have labeled data for the target domain. Therefore, the pivots selected using this approach are not guaranteed to demonstrate the same sentiment in the target domain as in the source domain. On the other hand, the pivot selection method proposed in this paper focuses on identifying a subset of features that are closely associated with both domains.

It is noteworthy that our proposed cross-domain word representation learning method (Section 3.2) *does not* assume any specific pivot/non-pivot selection method. Therefore, in principle, our proposed word representation learning method could be used with any of the previously proposed pivot selection methods. We defer a comprehensive evaluation of possible combinations of pivot selection methods and their effect on the proposed word representation learning method to future work.

3.5 Cross-Domain Sentiment Classification

As a concrete application of cross-domain word representations, we describe a method for learning a cross-domain sentiment classifier using the word representations learnt by the proposed method. Existing word representation learning methods that learn from only a single domain are typically evaluated for their accuracy in measuring semantic similarity between words, or by solving word analogy problems. Unfortunately, such gold standard datasets capturing cross-domain semantic variations of words are unavailable. Therefore, by applying the learnt word representations in a cross-domain sentiment classification task, we can conduct an indirect extrinsic evaluation.

The train data available for unsupervised cross-domain sentiment classification consists of unlabeled data for both the source and the target domains as well as labeled data for the source domain. We train a binary sentiment classifier using those train data, and apply it to classify sentiment of the target test data.

Unsupervised cross-domain sentiment classification is challenging due to two reasons: *feature-mismatch*, and *semantic variation*. First, the sets of features that occur in source and target domain documents are different. Therefore, a sentiment classifier trained using source domain labeled data is likely to encounter unseen features during test time. We refer to this as the feature-mismatch problem. Second, some of the features that occur in both domains will have different sentiments associated with them (e.g. *lightweight*). Therefore, a sentiment classifier trained using source domain labeled data is likely to incorrectly predict similar sentiment (as in the source) for such features. We call this the semantic variation problem. Next, we propose a method to overcome both problems using cross-domain word representations.

Let us assume that we are given a set $\{(\boldsymbol{x}_{\mathcal{S}}^{(i)}, y^{(i)})\}_{i=1}^{n}$ of n labeled reviews $\boldsymbol{x}_{\mathcal{S}}^{(i)}$ for the source domain $\mathcal{S}$. For simplicity, let us consider binary sentiment classification where each review $\boldsymbol{x}^{(i)}$ is labeled either as positive (i.e. $y^{(i)} = 1$) or negative (i.e. $y^{(i)} = -1$). Our cross-domain binary sentiment classification method can be easily extended to multi-class classification. First, we lemmatize each word in a source domain labeled review $\boldsymbol{x}_{\mathcal{S}}^{(i)}$, and extract unigrams and bigrams as features to represent $\boldsymbol{x}_{\mathcal{S}}^{(i)}$ by a binary-valued feature vector. Next, we train a binary linear classifier, $\boldsymbol{\theta}$, using those feature vectors. Any binary classification algorithm can be used for this purpose. We use $\boldsymbol{\theta}(z)$ to denote the weight learnt by the classifier for a feature z. In our experiments, we used l_2 regularized logistic regression.

At test time, we represent a test target review by a binary-valued vector $\boldsymbol{h}$ using a the set of unigrams and bigrams extracted from that review. Then, the activation score, $\psi(\boldsymbol{h})$, of $\boldsymbol{h}$ is defined

by:

$$\psi(\boldsymbol{h}) = \sum_{c \in \boldsymbol{h}} \sum_{c' \in \boldsymbol{\theta}} \boldsymbol{\theta}(c') f(\boldsymbol{c}'_{\mathcal{S}}, \boldsymbol{c}_{\mathcal{S}}) + \sum_{w \in \boldsymbol{h}} \sum_{w' \in \boldsymbol{\theta}} \boldsymbol{\theta}(w') f(\boldsymbol{w}'_{\mathcal{S}}, \boldsymbol{w}_{\mathcal{T}}) \tag{13}$$

Here, f is a similarity measure between two vectors. If $\psi(\boldsymbol{h}) > 0$, we classify $\boldsymbol{h}$ as positive, and negative otherwise. Eq. 13 measures the similarity between each feature in $\boldsymbol{h}$ against the features in the classification model $\boldsymbol{\theta}$. For pivots $c \in \boldsymbol{h}$, we use the the source domain representations to measure similarity, whereas for the (target-specific) non-pivots $w \in \boldsymbol{h}$, we use their target domain representations. We experimented with several popular similarity measures for f and found cosine similarity to perform consistently well. We can interpret Eq. 13 as a method for *expanding* a test target document using nearest neighbor features from the source domain labeled data. It is analogous to query expansion used in information retrieval to improve document recall (Fang, 2008). Alternatively, Eq. 13 can be seen as a linearly-weighted additive kernel function over two feature spaces.

4 Experiments and Results

For train and evaluation purposes, we use the Amazon product reviews collected by Blitzer et al. (2007) for the four product categories: books (**B**), DVDs (**D**), electronic items (**E**), and kitchen appliances (**K**). There are 1000 positive and 1000 negative sentiment labeled reviews for each domain. Moreover, each domain has on average $17,547$ unlabeled reviews. We use the standard split of 800 positive and 800 negative labeled reviews from each domain as training data, and the rest (200+200) for testing. For validation purposes we use *movie* (source) and *computer* (target) domains, which were also collected by Blitzer et al. (2007), but not part of the train/test domains.

Experiments conducted using this validation dataset revealed that the performance of the proposed method is relatively insensitive to the value of the regularization parameter $\lambda \in [10^{-3}, 10^{3}]$. For the non-pivot prediction task we generate positive and negative instances using the procedure described in Section 3.2. As a typical example, we have $88,494$ train instances from the books source domain and $141,756$ train instances from the target domain (1:5 ratio between positive and negative instances in each domain). The number of pivots and non-pivots are set to $N_{\mathcal{P}} = N_{\mathcal{S}} = N_{\mathcal{T}} = 500$.

In Figure 1, we compare the proposed method against two baselines (**NA**, **InDomain**), current state-of-the-art methods for unsupervised cross-domain sentiment classification (**SFA**, **SCL**), word representation learning (**GloVe**), and cross-domain similarity prediction (**CS**). The **NA** (no-adapt) lower baseline uses a classifier trained on source labeled data to classify target test data without any domain adaptation. The **InDomain** baseline is trained using the labeled data for the target domain, and simulates the performance we can expect to obtain if target domain labeled data were available. Spectral Feature Alignment (**SFA**) (Pan et al., 2010) and Structural Correspondence Learning (**SCL**) (Blitzer et al., 2007) are the state-of-the-art methods for cross-domain sentiment classification. However, those methods do not learn word representations.

We use Global Vector Prediction (**GloVe**) (Pennington et al., 2014), the current state-of-the-art word representation learning method, to learn word representations separately from the source and target domain unlabeled data, and use the learnt representations in Eq. 13 for sentiment classification. In contrast to the *joint* word representations learnt by the proposed method, **GloVe** simulates the level of performance we would obtain by learning representations *independently*. **CS** denotes the cross-domain vector prediction method proposed by Bollegala et al. (2014). Although **CS** can be used to learn a vector-space translation matrix, it *does not* learn word representations. Vertical bars represent the classification accuracies (i.e. percentage of the correctly classified test instances) obtained by a particular method on target domain's test data, and Clopper-Pearson 95% binomial confidence intervals are superimposed.

Differences in data pre-processing (tokenization/lemmatization), selection (train/test splits), feature representation (unigram/bigram), pivot selection (MI/frequency), and the binary classification algorithms used to train the final classifier make it difficult to directly compare results published in prior work. Therefore, we re-run the original algorithms on the same processed dataset under the same conditions such that any differences reported in Figure 1 can be directly attributable to the domain adaptation, or word-representation learning methods compared.

All methods use l_2 regularized logistic regression as the binary sentiment classifier, and the reg-

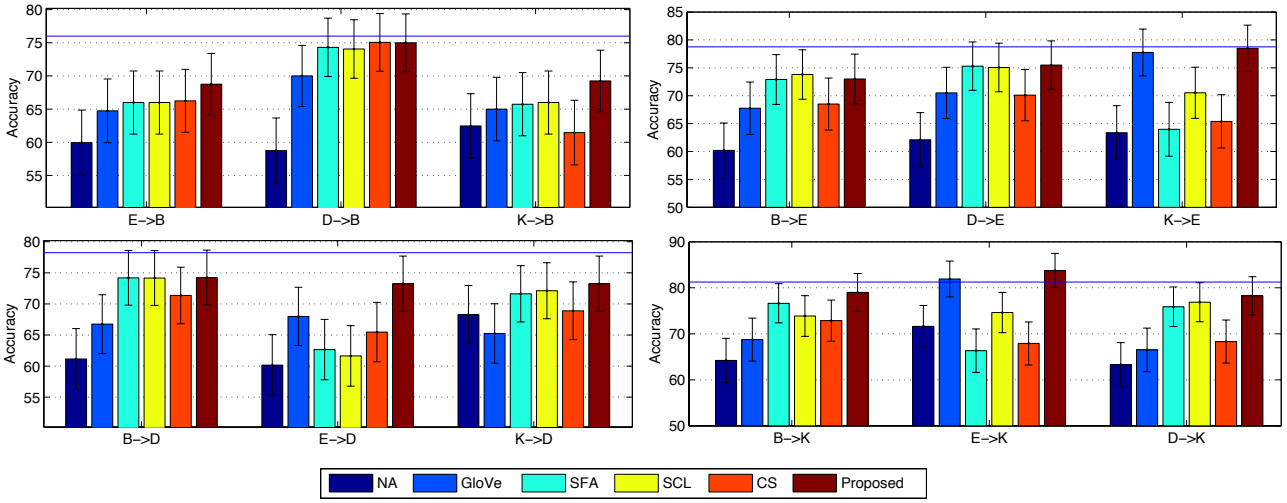

Figure 1: Accuracies obtained by different methods for each source-target pair in cross-domain sentiment classification.

ularization coefficients are set to their optimal values on the validation dataset. **SFA**, **SCL**, and **CS** use the same set of 500 pivots as used by the proposed method selected using NPMI (Section 3.4). Dimensionality n of the representation is set to 300 for both **GloVe** and the proposed method.

From Fig. 1 we see that the proposed method reports the highest classification accuracies in all 12 domain pairs. Overall, the improvements of the proposed method over **NA**, **GloVe**, and **CS** are statistically significant, and is comparable with **SFA**, and **SCL**. The proposed method's improvement over **CS** shows the importance of *predicting* word representations instead of *counting*. The improvement over **GloVe** shows that it is inadequate to simply apply existing word representation learning methods to learn independent word representations for the source and target domains.

We must consider the correspondences between the two domains as expressed by the pivots to jointly learn word representations. As shown in Fig. 2, the proposed method reports superior accuracies over **GloVe** across different dimensionalities. Moreover, we see that when the dimensionality of the representations increases, initially accuracies increase in both methods and saturates after $200 - 600$ dimensions. However, further increasing the dimensionality results in unstable and some what poor accuracies due to overfitting when training high-dimensional representations. Although our word representations learnt by the proposed method are not specific to sentiment classification, the fact that it clearly outperforms **SFA** and **SCL** in all domain pairs is encouraging, and implies the wider-applicability of the proposed method for domain adaptation tasks beyond sentiment classification.

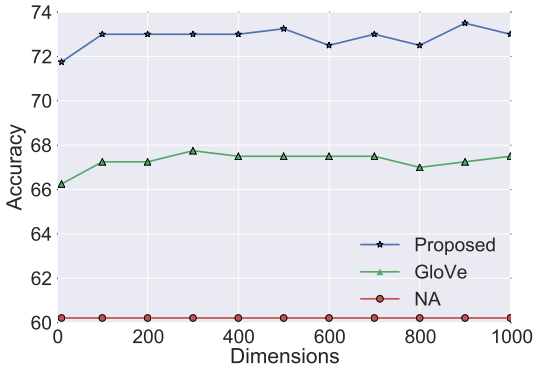

Figure 2: Accuracy vs. dimensionality of the representation.

5 Conclusion

We proposed an unsupervised method for learning cross-domain word representations using a given set of pivots and non-pivots selected from a source and a target domain. Moreover, we proposed a domain adaptation method using the learnt word representations.

Experimental results on a cross-domain sentiment classification task showed that the proposed method outperforms several competitive baselines and achieves best sentiment classification accuracies for all domain pairs. In future, we plan to apply the proposed method to other types of domain adaptation tasks such as cross-domain part-of-speech tagging, named entity recognition, and relation extraction.

Source code and pre-processed data etc. for this publication are publicly available[3].

[3] www.csc.liv.ac.uk/~danushka/prj/darep

References

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673 – 721.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proc. of ACL*, pages 238–247.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137 – 1155.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proc. of EMNLP*, pages 120 – 128.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. of ACL*, pages 440 – 447.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2011a. Relation adaptation: Learning to extract novel relations with minimum supervision. In *Proc. of IJCAI*, pages 2205 – 2210.

Danushka Bollegala, David Weir, and John Carroll. 2011b. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *ACL/HLT*, pages 132 – 141.

Danushka Bollegala, Mitsuru Kusumoto, Yuichi Yoshida, and Ken ichi Kawarabayashi. 2013a. Mining for analogous tuples from an entity-relation graph. In *Proc. of IJCAI*, pages 2064 – 2070.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2013b. Minimally supervised novel relation extraction using latent relational mapping. *IEEE Transactions on Knowledge and Data Engineering*, 25(2):419 – 432.

Danushka Bollegala, David Weir, and John Carroll. 2014. Learning to predict distributions of words across domains. In *Proc. of ACL*, pages 613 – 623.

Gerlof Bouma. 2009. Normalized (pointwsie) mutual information in collocation extraction. In *Proc. of GSCL*, pages 31 – 40.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuska. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493 – 2537.

Nguyen Tuan Duc, Danushka Bollegala, and Mitsuru Ishizuka. 2010. Using relational similarity between word pairs for latent relational search on the web. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 196 – 199.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121 – 2159, July.

Hui Fang. 2008. A re-examination of query expansion using lexical resources. In *Proc. of ACL*, pages 139–147.

Stefano Faralli and Roberto Navigli. 2012. A new minimally-supervised framework for domain word sense disambiguation. In *EMNLP*, pages 1411 – 1422.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proc. of ICML*.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proc. of ICML*.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual distributed representations without word alignment. In *Proc. of ICLR*.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proc. of ACL*, pages 873 – 882.

Jing Jiang and ChengXiang Zhai. 2007a. Instance weighting for domain adaptation in nlp. In *ACL 2007*, pages 264 – 271.

Jing Jiang and ChengXiang Zhai. 2007b. A two-stage approach to domain adaptation for statistical classifiers. In *CIKM 2007*, pages 401–410.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proc. of COLING*, pages 1459 – 1474.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Proc. of NAACL/HLT*, pages 28 – 36.

Tomas Mikolov, Kai Chen, and Jeffrey Dean. 2013a. Efficient estimation of word representation in vector space. *CoRR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proc. of NIPS*, pages 3111 – 3119.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continous space word representations. In *NAACL'13*, pages 746 – 751.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proc. of ACL-HLT*, pages 236 – 244.

Andriy Mnih and Geoffrey E. Hinton. 2008. A scalable hierarchical distributed language model. In *Proc. of NIPS*, pages 1081–1088.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Proc. of NIPS*.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proc. of EMNLP*, pages 1059–1069.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proc. of WWW*, pages 751 – 760.

Jeffery Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: global vectors for word representation. In *Proc. of EMNLP*.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Proc. of HLT-NAACL*, pages 109 – 117.

Michael Roth and Kristian Woodsend. 2014. Composition of word representations improves semantic role labelling. In *Proc. of EMNLP*, pages 407–413.

Tobias Schnabel and Hinrich Schütze. 2013. Towards robust cross-domain domain adaptation for part-of-speech tagging. In *Proc. of IJCNLP*, pages 198 – 206.

Richard Socher, Cliff Chiung-Yu Lin, Andrew Ng, and Chris Manning. 2011a. Parsing natural scenes and natural language with recursive neural networks. In *ICML'11*.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011b. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proc. of EMNLP*, pages 151–161.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Aritificial Intelligence Research*, 37:141 – 188.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proc. of EMNLP'13*, pages 1393 – 1398.