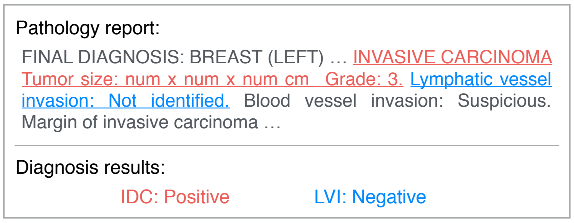
Aspect-augmented Adversarial Networks for Domain Adaptation

Yuan Zhang, Regina Barzilay, and Tommi Jaakkola
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{yuanzh, regina, tommi}@csail.mit.edu

Abstract

We introduce a neural method for transfer learning between two (source and target) classification tasks or aspects over the same domain. Instead of target labels, we assume a few keywords pertaining to source and target aspects indicating sentence relevance rather than document class labels. Documents are encoded by learning to embed and softly select relevant sentences in an aspect-dependent manner. A shared classifier is trained on the source encoded documents and labels, and applied to target encoded documents. We ensure transfer through aspect-adversarial training so that encoded documents are, as sets, aspect-invariant. Experimental results demonstrate that our approach outperforms different baselines and model variants on two datasets, yielding an improvement of 24% on a pathology dataset and 5% on a review dataset.[1]



Pathology report:

FINAL DIAGNOSIS: BREAST (LEFT) … INVASIVE CARCINOMA Tumor size: num x num x num cm Grade: 3. Lymphatic vessel invasion: Not identified. Blood vessel invasion: Suspicious. Margin of invasive carcinoma …

Diagnosis results:

IDC: Positive LVI: Negative

Figure 1: A snippet of a breast pathology report with diagnosis results for two types of disease. Evidence for both results is in red and blue, respectively.

1 Introduction

Deep learning methods are highly effective when they can be trained with large amounts of labeled training data in the domain of interest. While such data are not always available in real applications, it is nevertheless often possible to find labeled data in another related domain or for another related task. Considerable effort has gone into designing domain transfer algorithms that leverage such related data (Glorot et al., 2011; Chen et al., 2012; Zhou et al., 2016). In a typical case, the related domain involves the same classification task (e.g., sentiment analysis) but over different types of examples (e.g., hotel vs restaurant reviews). Labeled training data are available only in the source domain (e.g., hotel reviews) while the task is to provide an effective method for the target domain (e.g., restaurant reviews) without any additional labeled examples.

In this paper we are primarily interested in transfer between two classification tasks over the same domain, i.e., over the same set of examples. We call this "aspect transfer" as the two classification tasks can be thought to pertain to different aspects of the same examples. For example, the target goal may be to classify pathology reports (shown in Figure 1) for the presence of lymph invasion but the available training data involve only annotations for carcinoma in the same reports. Existing domain adaptation methods do not directly solve this aspect transfer problem because input examples are the same across the two tasks. Since there are no labels available for the target aspect, we must learn to properly relate the two tasks. In particular, we bring in auxiliary data to help connect the tasks.

[1] The code is available at https://github.com/yuanzh/aspect_adversarial.

Our approach builds on relevance annotations of sentences which are considerably easier to obtain than actual class labels. Relevance merely indicates a possibility that the answer could be found in a sentence, not what the answer is. One can often write simple keyword rules that identify sentence relevance to a particular aspect (task) through representative terms, e.g., specific hormonal markers in the context of pathology reports. We can also use keywords of other irrelevant aspects to indicate absence of relevance. Annotations of this kind can be readily provided by domain experts, or extracted from medical literature such as codex rules in pathology (Pantanowitz et al., 2008). We therefore assume a small number of relevance annotations pertaining to both source and target aspects as a form of weak supervision. These annotations permit us to learn how to encode the examples (e.g., pathology reports) from the point of view of the desired task. Specifically, differential encodings of the same report in our approach arise from softly selecting aspect-relevant sentences from the report.

Our relevance driven encoding returns the aspect-transfer problem closer to the realm of standard domain adaption. We employ a shared end classifier between the tasks but it is exercised differently due to aspect-driven encoding of examples. The two domains as in standard domain adaption are therefore induced by different ways of interpreting the same example in our case. These interpretations are themselves learned based on relevance feedback, thus naturally pulled apart. To ensure that the classifier can be adjusted only based on the source class labels and still reasonably applied to the target encodings, we must align the two sets of encoded examples. Note that this alignment or invariance is enforced on the level of sets, not individual examples or reports; encoding of any specific report should remain substantially different for label prediction. To learn the invariance, we introduce an adversarial domain classifier analogously to recent successful use of adversarial training in computer vision (Ganin and Lempitsky, 2014). The role of the adversarial domain classifier is to learn to distinguish between the two types of encodings, establishing invariance (as sets) when it fails. All the three components in our approach, 1) aspect-driven encoding, 2) classification of source labels, and 3) domain adversary, are trained jointly (concurrently) to complement and balance each other.

Adversarial training of domain and end classifiers can be challenging to stabilize. In our setting, sentences are encoded with a shared convolutional model, weighted by predicted aspect relevance, and then combined into aspect-driven document representations. Feedback from adversarial training can be an unstable guide for how the sentences should be encoded in the first place. To this end, we incorporate an additional word-level autoencoder reconstruction loss to ground the convolutional processing of sentences. We empirically demonstrate that this additional objective yields richer and more diversified feature representations, improving transfer.

We evaluate our approach on pathology reports (aspect transfer) as well as on a more standard review dataset (domain adaptation). On the pathology dataset, we explore cross-aspect transfer across different types of breast disease. Specifically, we test on six adaptation tasks, consistently outperforming all other baselines. Overall, our full model achieves 24% and 12.8% absolute improvement arising from aspect-driven encoding and adversarial training, respectively. Moreover, our unsupervised adaptation method is only 2.8% behind the accuracy of a supervised target model. On the review dataset, we test adaptation from hotel to restaurant reviews. Our model outperforms the marginalized denoising autoencoder (Chen et al., 2012) by 5%. Finally, we examine and illustrate the impact of individual components on the resulting performance.

2 Related Work

Domain Adaptation for Deep Learning Existing approaches commonly induce abstract representations without pulling apart different aspects in the same example, and therefore are likely to fail on the aspect transfer problem. The majority of these prior methods propose to first learn a task-independent representation, and then train a label predictor (e.g. SVM) on this representation in a separate step. For example, earlier researches employ a shared autoencoder (Glorot et al., 2011; Chopra et al., 2013) to learn cross-domain representation. Chen et al. (2012) further improve and stabilize the representation learning by utilizing marginalized denoising autoencoders. Later, Zhou et al. (2016) propose to minimize domain-shift of the autoencoder in a linear data combination manner. Some other work has focused on learning transferable representations in an end-to-end fashion. Examples include using transduction learning for object recognition (Sener et al., 2016) and using residual transfer networks for image classification (Long et al., 2016). In contrast, we use adversarial training to encourage learning domain-invariant features in a more explicit way. Our approach offers another two advantages over prior work. First, we jointly optimize features with the final classification task while much previous work only learns task-independent features using autoencoders. Second, our model can handle traditional domain transfer as well as aspect transfer, while previous methods can only handle the former scenario.

Adversarial Learning in Vision and NLP Our approach closely relates to the idea of domain-adversarial training. Adversarial networks have originally been developed for image generation (Goodfellow et al., 2014; Makhzani et al., 2015; Springenberg, 2015; Radford et al., 2015; Taigman et al., 2016), and later applied to domain adaption in computer vision (Ganin and Lempitsky, 2014; Ganin et al., 2015; Bousmalis et al., 2016; Tzeng et al., 2014) and speech recognition (Shinohara, 2016). The core idea of these approaches is to promote the emergence of invariant image features by optimizing the feature extractor as an adversary against the domain classifier. While Ganin et al. (2015) also apply this idea to sentiment analysis, their practical gains have remained limited.

Our approach presents two main departures. In computer vision, adversarial learning has been used for transferring across domains, while our method can also handle aspect transfer. In addition, we introduce reconstruction loss which results in more robust adversarial training. We believe that this formulation will benefit other applications of adversarial training, beyond the ones described in this paper.

Semi-supervised Learning with Keywords In our work, we use a small set of keywords as a source of weak supervision for aspect-relevance scoring. This relates to prior work on utilizing prototypes and seed words in semi-supervised learning (Haghighi and Klein, 2006; Grenager et al., 2005; Chang et al., 2007; Mann and McCallum, 2008; Jagarlamudi et al., 2012; Li et al., 2012; Eisenstein, 2017). All these prior approaches utilize prototype annotations primarily targeting for model bootstrapping but not for learning representations. In contrast, our model uses provided keywords to learn aspect-driven encoding of input examples.

Attention Mechanism in NLP One may view our aspect-relevance scorer as a sentence-level "semi-supervised attention", where relevant sentences receive more attention during feature extraction. While traditional attention-based models typically induce attention in an unsupervised manner, they have to rely on a large amount of labeled data for the target task (Bahdanau et al., 2014; Rush et al., 2015; Chen et al., 2015; Cheng et al., 2016; Xu et al., 2015; Xu and Saenko, 2015; Yang et al., 2015; Martins and Astudillo, 2016; Lei et al., 2016). Unlike them, we assume no label annotations in the target domain. Some other researches have focused on utilizing human-provided rationales as "supervised attention" to improve prediction (Zaidan et al., 2007; Marshall et al., 2015; Zhang et al., 2016; Brun et al., 2016). In contrast, our model only assumes access to a small set of keywords as a source of weak supervision. Moreover, all these prior approaches focus on in-domain classification. In this paper, however, we study the task in the context of domain adaptation.

3 Methods

We formalize here the aspect transfer problem between the source and target classification tasks over the same set of examples (here documents, e.g., pathology reports). Class labels are available only for the source task, and the goal is to solve the target classification task. While we develop our method under the assumption that the examples in the two tasks are the same (as an extreme case), this is not a requirement for our method and it will work fine in more traditional domain adaptation settings as well, which we demonstrate.

Let $\mathbf{d} = \{\mathbf{s}_i\}_{i=1}^{|\mathbf{d}|}$ be a document that consists of a sequence of $|\mathbf{d}|$ sentences. Each sentence is a sequence of words, namely $\mathbf{s}_i = \{\mathbf{x}_{i,j}\}_{j=1}^{|\mathbf{s}_i|}$, where $\mathbf{x}_{i,j} \in \mathbb{R}^d$ denotes the vector representation of the j-th word in the i-th sentence. Given a document $\mathbf{d}$

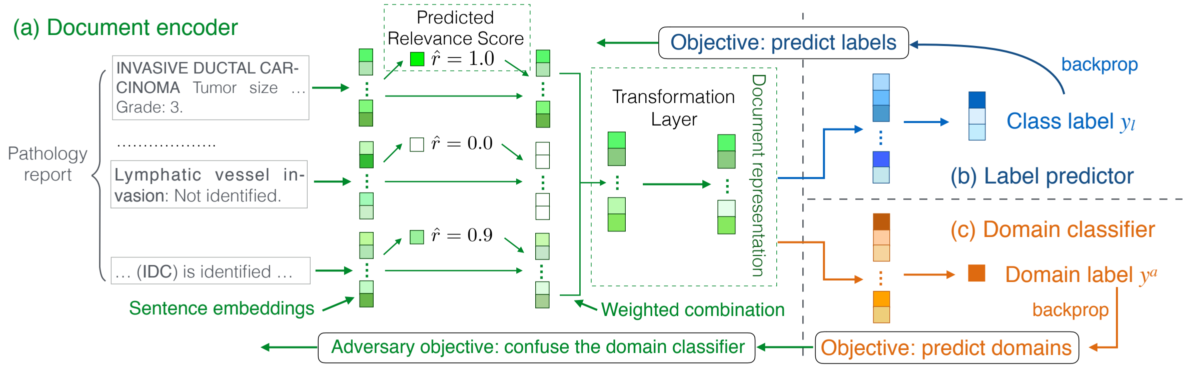


Figure 2: Aspect-augmented adversarial network for domain adaptation. The model is composed of (a) an aspect-driven document encoder, (b) a label predictor and (c) a domain classifier.

we wish to predict the corresponding class label y (e.g., $y \in \{-1, 1\}$) which varies for the same document depending on which aspect (source, target) we are interested in. We assume that the set of possible labels are the same across tasks. We use $y^s_{l;k}$ to denote the k-th coordinate of a one-hot vector indicating the correct source label for document $\mathbf{d}_l$.

Beyond labeled documents for the source task $\{\mathbf{d}_l, y^s_l\}_{l \in L}$, and shared unlabeled documents for source and target tasks $\{\mathbf{d}_l\}_{l \in U}$, we assume further that we have relevance scores pertaining to each aspect. The relevance is given per sentence, for some subset of sentences across the documents, and indicates the possibility that the answer for that document would be found in the sentence but without indicating which way the answer goes. Relevance is always task (aspect) dependent yet often easy to provide with simple keyword rules. We use $r^a_i \in \{0, 1\}$ to denote the given relevance label pertaining to aspect a for sentence $\mathbf{s}_i$. Specifically, if sentence $\mathbf{s}_i$ has a relevance label, then $r^a_i = 1$ when the sentence contains any keywords pertaining to aspect a and $r^a_i = 0$ if it has any keywords of other aspects. Separate subsets of relevance labels are available for each task as the keywords differ. Let $R = \{(a, l, i)\}$ denote the index set of relevance labels such that if $(a, l, i) \in R$ then relevance label $r^a_{l,i}$ is available for aspect a and the i^{th} sentence in document $\mathbf{d}_l$.

3.1 Our Approach

Figure 2 outlines the overall model. Each sentence is first encoded into a vector using a shared convolutional model. We ground this convolutional model by including a reconstruction step for each word based on the internal state centered at the same position. The sentence vectors are then passed on to a single hidden layer network, a separate network for each aspect with a shared hidden layer, to determine whether the sentences are relevant for the chosen aspect. Our relevance predictors are non-negative regression methods as relevance varies more on a linear rather than binary scale. The predicted relevance scores are used to construct document vectors by taking relevance-weighted combinations of the associated sentence vectors. Thus the document vector is always aspect-dependent due to the chosen relevance weights. The constrained manner in which these document vectors arise from sentence vectors means that they will retain explicit information about the aspect they were based on. Such explicit cues are not helpful in our setting: the end classifier, trained only on source labels, would unnecessarily rely on cues present only in source-aspect encodings. To remove those cues, we introduce an additional linear transformation layer after the initial document encoding.

During training, the resulting adjusted document vectors are used by two classifiers, each involving one hidden layer. The primary end classifier aims to predict the source labels (when available), while the domain classifier determines whether the document vector pertains to the source or target aspect (i.e., label that we know by construction). The two classifiers involve separate training losses that interact only in terms of the document representation. Specifically, the training signal from the primary classifier is used to co-operatively adjust the document representation whereas the gradient from the

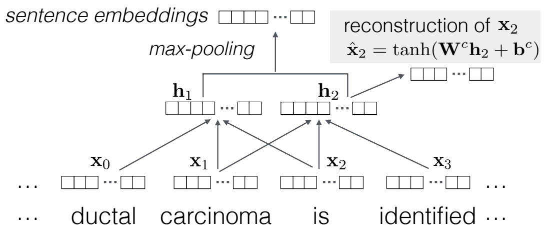

sentence embeddings
max-pooling
reconstruction of $\mathbf{x}_2$
$\hat{\mathbf{x}}_2 = \tanh(\mathbf{W}^c\mathbf{h}_2 + \mathbf{b}^c)$
$\mathbf{h}_1$
$\mathbf{h}_2$
$\mathbf{x}_0$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
… ductal carcinoma is identified …

Figure 3: Illustration of the convolutional model and the reconstruction of word embeddings from the associated convolutional layer.

domain classifier (the adversary) is reversed therefore encouraging representations that make it fail.

The four training losses pertaining to word reconstruction, relevance labels, source class labels, and domain labels are used concurrently in our adversarial training scheme to adjust the model parameters. At the conclusion of training, we expect that the primary classify is able to predict the source labels while appearing to be exercised in a domain invariant manner, enabling transfer to the target task.

3.2 Components in detail

Sentence embedding We apply a convolutional model illustrated in Figure 3 to each sentence $\mathbf{s}_i$ to obtain sentence-level vector embeddings $\mathbf{x}_i^{sen}$. The use of RNNs or bi-LSTMs would result in more flexible sentence embeddings but based on our initial experiments, we did not observe any significant gains over the simpler CNNs.

We introduce an additional word-level reconstruction step in the convolutional model to further ground the resulting sentence embeddings. The purpose of this reconstruction step is to balance adversarial training signals propagating back from the domain classifier. Specifically, it forces the sentence encoder to keep rich word-level information in contrast to adversarial training that seeks to eliminate task/aspect specific features. We provide an empirical analysis of the impact of this reconstruction in the experiment section (Section 6).

More concretely, we reconstruct word embedding from the corresponding convolutional layer, as shown in Figure 3. Let $\mathbf{h}_{i,j}$ be the convolutional output when $\mathbf{x}_{i,j}$ is at the center of the window. We reconstruct $\mathbf{x}_{i,j}$ by

$$\hat{\mathbf{x}}_{i,j} = \tanh(\mathbf{W}^c\mathbf{h}_{i,j} + \mathbf{b}^c) \qquad (1)$$

where $\mathbf{W}^c$ and $\mathbf{b}^c$ are parameters of the reconstruction layer. The loss associated with the reconstruction for document $\mathbf{d}$ is

$$\mathcal{L}^{rec}(\mathbf{d}) = \frac{1}{n}\sum_{i,j} ||\hat{\mathbf{x}}_{i,j} - \tanh(\mathbf{x}_{i,j})||_2^2 \qquad (2)$$

where n is the number of tokens in the document and indexes i, j identify the sentence and word, respectively. The overall loss $\mathcal{L}^{rec}$ is obtained by summing over all labeled/unlabeled documents.

Relevance prediction We use a small set of keyword rules to generate binary relevance labels, both positive ($r = 1$) and negative relevance ($r = 0$). These labels represent the only supervision available to predict relevance. The prediction is made on the basis of the sentence vector $\mathbf{x}_i^{sen}$ passed through a feed-forward network with a ReLU output unit. The network has a single shared hidden layer and a separate output layer for each aspect. Note that our relevance prediction network is trained as a regression model even though the available labels are binary.

Given relevance labels indexed by $R = \{(a, l, i)\}$, we minimize

$$\mathcal{L}^{rel} = \sum_{(a,l,i)\in R} \left(r_{l,i}^a - \hat{r}_{l,i}^a\right)^2 \qquad (3)$$

where $\hat{r}_{l,i}^a$ is the predicted (non-negative) relevance score pertaining to aspect a for the i^{th} sentence in document $\mathbf{d}_l$, as shown in the left part of Figure 2. $r_{l,i}^a$, defined earlier, is the given binary (0/1) relevance label.

Document encoding The initial vector representation for each document such as $\mathbf{d}_l$ is obtained as a relevance weighted combination of the associated sentence vectors, i.e.,

$$\mathbf{x}_l^{doc,a} = \frac{\sum_i \hat{r}_{l,i}^a \cdot \mathbf{x}_{l,i}^{sen}}{\sum_i \hat{r}_{l,i}^a} \qquad (4)$$

The resulting vector selectively encodes information from the sentences based on relevance to the focal aspect.

Transformation layer We add a transformation layer to help map the initial document vectors $\mathbf{x}_l^{doc,a}$ to their domain invariant (as a set) versions. Specifically, the transformed representation is given by

$\mathbf{x}_l^{tr,a} = \mathbf{W}^{tr}\mathbf{x}_l^{doc,a}$. The transformation has to be strongly regularized lest the gradient from the adversary would wipe out all the document signal. We add the following regularization term

$$\Omega^{tr} = \lambda^{tr}||\mathbf{W}^{tr} - \mathbf{I}||_F^2 \quad (5)$$

to discourage significant deviation away from identity $\mathbf{I}$. λ^{tr} is a regularization parameter that has to be set separately based on validation performance. We show an empirical analysis of the impact of this transformation layer in Section 6.

Primary label classifier As shown in the top-right part of Figure 2, the classifier takes in the adjusted document representation as an input and predicts a probability distribution over the possible class labels. The classifier is a feed-forward network with a single hidden layer using ReLU activations and a softmax output layer over the possible class labels. Note that the classifier operates the same regardless of the aspect relative to which the document was encoded. It must therefore be co-operatively learned together with the encodings.

Let $\hat{p}_{l;k}$ denote the predicted probability of class k for document $\mathbf{d}_l$ when the document is encoded from the point of view of the source aspect. Recall that $[y^s_{l;1}, \ldots, y^s_{l;m}]$ is a one-hot vector for the correct (given) source class label for document $\mathbf{d}_l$, hence also a distribution. We use the cross-entropy loss for the label classifier

$$\mathcal{L}^{lab} = \sum_{l \in L} \left[-\sum_{k=1}^{m} y^s_{l;k} \log \hat{p}_{l;k} \right] \quad (6)$$

Domain classifier As shown in the bottom-right part of Figure 2, the domain classifier functions as an adversary to ensure that the documents encoded with respect to the source and target aspects look the same as sets of examples. The invariance is achieved when the domain classifier (as the adversary) fails to distinguish between the two. Structurally, the domain classifier is a feed-forward network with a single ReLU hidden layer and a softmax output layer over the two aspect labels.

Let $y^a = [y^a_1, y^a_2]$ denote the one-hot domain label vector for aspect $a \in \{s, t\}$. In other words, $y^s = [1, 0]$ and $y^t = [0, 1]$. We use $\hat{q}_k(\mathbf{x}_l^{tr,a})$ as the predicted probability that the domain label is k when the domain classifier receives $\mathbf{x}_l^{tr,a}$ as the input. The domain classifier is trained to minimize

$$\mathcal{L}^{dom} = \sum_{l \in L \cup U} \sum_{a \in \{s,t\}} \left[-\sum_{k=1}^{2} y^a_k \log \hat{q}_k(\mathbf{x}_l^{tr,a}) \right] \quad (7)$$

Dataset		#Labeled	#Unlabeled
Pathology	DCIS	23.8k	96.6k
	LCIS	10.7k	
	IDC	22.9k	
	ALH	9.2k	
Review	Hotel	100k	100k
	Restaurant	-	200k

Table 1: Statistics of the pathology reports dataset and the reviews dataset that we use for training. Our model utilizes both labeled and unlabeled data.

3.3 Joint learning

We combine the individual component losses into an overall objective function

$$\mathcal{L}^{all} = \mathcal{L}^{rec} + \mathcal{L}^{rel} + \Omega^{tr} + \mathcal{L}^{lab} - \rho\mathcal{L}^{dom} \quad (8)$$

which is minimized with respect to the model parameters except for the adversary (domain classifier). The adversary is maximizing the same objective with respect to its own parameters. The last term $-\rho\mathcal{L}^{dom}$ corresponds to the objective of failing the domain classifier. The proportionality constant ρ controls the impact of gradients from the adversary on the document representation; the adversary itself is always directly minimizing $\mathcal{L}^{dom}$.

All the parameters are optimized jointly using standard backpropagation (concurrent for the adversary). Each mini-batch is balanced by aspect, half coming from the source, the other half from the target. All the loss functions except $\mathcal{L}^{lab}$ make use of both labeled and unlabeled documents. It would be straightforward to add a loss term also for target labels if they are available.

4 Experimental Setup

Pathology dataset This dataset contains 96.6k breast pathology reports collected from three hospitals (Yala et al., 2016). A portion of this dataset is manually annotated with 20 categorical values,

ASPECT	KEYWORDS
IDC	IDC, Invasive Ductal Carcinoma
ALH	ALH, Atypical Lobular Hyperplasia

Table 2: Examples of aspects and their corresponding keywords (case insensitive) in the pathology dataset.

representing various aspects of breast disease. In our experiments, we focus on four aspects related to carcinomas and atypias: Ductal Carcinoma In-Situ (DCIS), Lobular Carcinoma In-Situ (LCIS), Invasive Ductal Carcinoma (IDC) and Atypical Lobular Hyperplasia (ALH). Each aspect is annotated using binary labels. We use 500 held out reports as our test set and use the rest labeled data as our training set: 23.8k reports for DCIS, 10.7k for LCIS, 22.9k for IDC, and 9.2k for ALH. Table 1 summarizes statistics of the dataset.

We explore the adaptation problem from one aspect to the other. For example, we want to train a model on annotations of DCIS and apply it on LCIS. For each aspect, we use up to three common names as a source of supervision for learning the relevance scorer, as illustrated in Table 2. Note that the provided list is by no means exhaustive. In fact Buckley et al. (2012) provide example of 60 different verbalizations of LCIS, not counting negations.

Review dataset Our second experiment is based on a domain transfer of sentiment classification. As the source domain, we use the hotel review dataset introduced in previous work (Wang et al., 2010; Wang et al., 2011). For the target domain, we use the restaurant review dataset from Yelp.[2] Both datasets have ratings on a scale of 1 to 5 stars. Following previous work (Blitzer et al., 2007), we label reviews with ratings > 3 as positive and those with ratings < 3 as negative, and we discard the rest. The hotel dataset includes a total of around 200k reviews collected from TripAdvisor,[3] so we split 100k as labeled and the other 100k as unlabeled data. We randomly select 200k restaurant reviews as the unlabeled data in the target domain. Our testing set consists of 2k reviews. Table 1 summarizes the statistics of the review dataset.

METHOD	SOURCE		TARGET	
	Label	Unlabel	Label	Unlabel
SVM	✓	×	×	×
SourceOnly	✓	✓	×	×
mSDA	✓	✓	×	✓
Ours-NA	✓	✓	×	✓
Ours-NR	✓	✓	×	✓
In-Domain	×	×	✓	×
Ours-Full	✓	✓	×	✓

Table 3: Usage of labeled and unlabeled data in each domain by our model and other baseline methods.

The hotel reviews naturally have ratings for six aspects, including *value*, *room* quality, *checkin* service, room *service*, *cleanliness* and *location*. We use the first five aspects because the sixth aspect *location* has positive labels for over 95% of the reviews and thus the trained model will suffer from the lack of negative examples. The restaurant reviews, however, only have single ratings for an *overall* impression. Therefore, we explore the task of adaptation from each of the five hotel aspects to the restaurant domain. The hotel reviews dataset also provides a total of 290 keywords for different aspects that are generated by the bootstrapping method used in (Wang et al., 2010). We use those keywords as supervision for learning the relevance scorer.

Baselines We first compare against a linear **SVM** trained on the raw bag-of-words representation of labeled data in source. Second, we compare against our **SourceOnly** model that assumes no target domain data or keywords. It thus has no adversarial training or target aspect-relevance scoring. Next we compare with marginalized Stacked Denoising Autoencoders (**mSDA**) (Chen et al., 2012), a domain adaptation algorithm that outperforms both prior deep learning and shallow learning approaches.[4] We also compare against **Ours-NA** and **Ours-NR** that are our model variants without adversarial training and without aspect-relevance scoring respectively. Finally we include supervised models trained on the full set of **In-Domain** annotations as the performance upper bound. Table 3 summarizes

[2] `https://www.yelp.com/dataset_challenge`

[3] `https://www.tripadvisor.com/`

[4] We use the publicly available implementation provided by the authors at `http://www.cse.wustl.edu/~mchen/code/mSDA.tar`. We use the hyper-parameters from the authors and their models have more parameters than ours.

Domain		SVM	Source Only	mSDA	Ours-NA	Ours-NR	Ours-Full	In-Domain
Source	Target							
LCIS	DCIS	45.8	25.2	45.0	81.2	50.0	**93.0**	96.2
DCIS	LCIS	73.8	75.4	76.2	89.0	81.2	**95.2**	97.8
DCIS	IDC	94.0	77.4	94.0	92.4	93.8	**95.4**	96.8
IDC	DCIS	71.8	62.4	73.0	87.6	81.4	**94.8**	96.2
ALH	LCIS	54.4	46.4	54.2	84.8	52.4	**93.2**	97.8
LCIS	ALH	59.0	51.6	60.4	52.6	60.0	**92.8**	96.8
Average		66.5	56.4	67.1	81.3	69.8	**94.1**	96.9

Table 4: **Pathology:** Classification accuracy (%) of different approaches on the pathology reports dataset, including the results of six adaptation scenarios from four different aspects (IDC, ALH, DCIS and LCIS) in breast cancer pathology reports. "mSDA" indicates the marginalized denoising autoencoder in (Chen et al., 2012). "Ours-NA" and "Ours-NR" corresponds to our model without the adversarial training and the aspect-relevance scoring component, respectively. We also include in the last column the in-domain supervised training results of our model as the performance upper bound. Boldface numbers indicate the best accuracy for each testing scenario.

the usage of labeled and unlabeled data in each domain by our model (Ours-Full) and different baselines. Note that our model assumes the same set of data as Ours-NA, Ours-NR and mSDA methods.

Implementation details Following prior work (Ganin and Lempitsky, 2014), we gradually increase the adversarial strength ρ and decay the learning rate during training. We also apply batch normalization (Ioffe and Szegedy, 2015) on the sentence encoder and apply dropout with ratio 0.2 on word embeddings and each hidden layer activation. We set the hidden layer size to 150 and pick the transformation regularization weight $\lambda^t = 0.1$ for the pathology dataset and $\lambda^t = 10.0$ for the review dataset.

5 Main Results

Table 4 summarizes the classification accuracy of different methods on the pathology dataset, including the results of six adaptation tasks. Our full model (Ours-Full) consistently achieves the best performance on each task compared with other baselines and model variants. It is not surprising that SVM and mSDA perform poorly on this dataset because they only predict labels based on an overall feature representation of the input, and do not utilize weak supervision provided by aspect-specific keywords. As a reference, we also provide a performance upper bound by training our model on the full labeled set in the target domain, denoted as In-Domain in the last column of Table 4. On average, the accuracy of our model is only 2.8% behind this upper bound.

Table 5 shows the adaptation results from each aspect in the hotel reviews to the overall ratings of restaurant reviews. Ours-Full and Ours-NR are the two best performing systems on this review dataset, attaining around 5% improvement over the mSDA baseline. Below, we summarize our findings when comparing the full model with the two model variants Ours-NA and Ours-NR.

Impact of adversarial training We first focus on comparisons between Ours-Full and Ours-NA. The only difference between the two models is that Ours-NA has no adversarial training. On the pathology dataset, our model significantly outperforms Ours-NA, yielding a 12.8% absolute average gain (see Table 4). On the review dataset, our model obtains 2.5% average improvement over Our-NA. As shown in Table 5, the gains are more significant when training on room quality and check-in service aspects, reaching 6.9% and 4.5%, respectively.

Impact of relevance scoring As shown in Table 4, the relevance scoring component plays a crucial role in classification on the pathology dataset. Our model achieves more than 24% improvement over Ours-NR. This is because in general aspects have zero correlations to each other in pathology reports. Therefore, it is essential for the model to have the capacity of distinguishing across different

DOMAIN SOURCE	 TARGET	SVM	Source Only	mSDA	Ours-NA	Ours-NR	Ours-Full	In-Domain
Value	Restaurant Overall	82.2	87.4	84.7	87.1	**91.1**	89.6	93.4
Room		75.6	79.3	80.3	79.7	86.1	**86.6**	
Checkin		77.8	83.0	81.0	80.9	**87.2**	85.4	
Service		82.2	88.0	83.8	88.8	87.9	**89.1**	
Cleanliness		77.9	83.2	78.4	83.1	**84.5**	81.4	
AVERAGE		79.1	84.2	81.6	83.9	**87.3**	86.4	93.4

Table 5: **Review:** Classification accuracy (%) of different approaches on the reviews dataset. Columns have the same meaning as in Table 4. Boldface numbers indicate the best accuracy for each testing scenario.

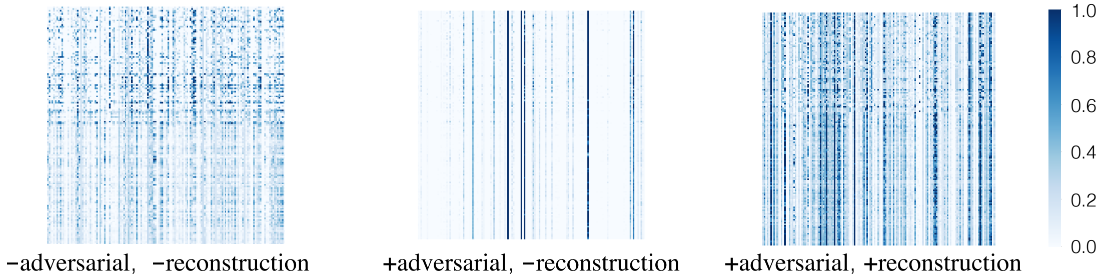


Figure 4: Heat map of 150×150 matrices. Each row corresponds to the vector representation of a document that comes from either the source domain (top half) or the target domain (bottom half). Models are trained on the review dataset when room quality is the source aspect.

aspects in order to succeed in this task.

On the review dataset, however, we observe that relevance scoring has no significant impact on performance. On average, Ours-NR actually outperforms Ours-Full by 0.9%. This observation can be explained by the fact that different aspects in hotel reviews are highly correlated to each other. For example, the correlation between room quality and cleanliness is 0.81, much higher than aspect correlations in the pathology dataset. In other words, the sentiment is typically consistent across all sentences in a review, so that selecting aspect-specific sentences becomes unnecessary. Moreover, our supervision for the relevance scorer is weak and noisy because the aspect keywords are obtained in a semi-automatic way. Therefore, it is not surprising that Ours-NR sometimes delivers a better classification accuracy than Ours-Full.

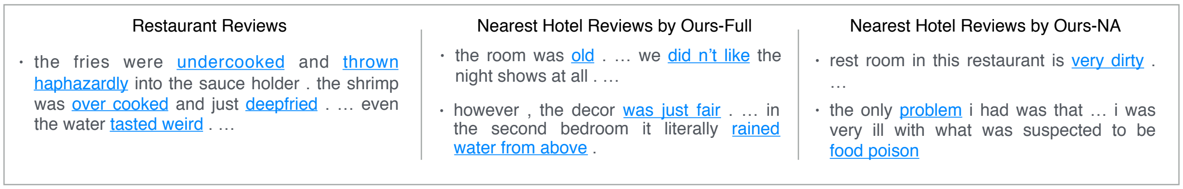
6 Analysis

DATASET	Ours-Full		Ours-NA	
	-REC.	+REC.	-REC.	+REC.
PATHOLOGY	89.5	94.1	78.6	81.3
REVIEW	80.8	86.4	85.0	83.9

Table 6: Impact of adding the reconstruction component in the model, measured by the average accuracy on each dataset. +REC. and -REC. denote the presence and absence of the reconstruction loss, respectively.

Impact of the reconstruction loss Table 6 summarizes the impact of the reconstruction loss on the model performance. For our full model (Ours-Full), adding the reconstruction loss yields an average of 4.6% gain on the pathology dataset and 5.2% on the review dataset.

To analyze the reasons behind this difference, consider Figure 4 that shows the heat maps of the learned document representations on the review dataset. The top half of the matrices corresponds to input documents from the source domain and the bottom half corresponds to the target domain. Unlike the first matrix, the other two matrices has no significant difference between the two halves, in-



Restaurant Reviews

- the fries were undercooked and thrown haphazardly into the sauce holder . the shrimp was over cooked and just deepfried . … even the water tasted weird . …

Nearest Hotel Reviews by Ours-Full

- the room was old . … we did n't like the night shows at all . …
- however , the decor was just fair . … in the second bedroom it literally rained water from above .

Nearest Hotel Reviews by Ours-NA

- rest room in this restaurant is very dirty . …
- the only problem i had was that … i was very ill with what was suspected to be food poison

Figure 5: Examples of restaurant reviews and their nearest neighboring hotel reviews induced by different models (column 2 and 3). We use room quality as the source aspect. The sentiment phrases of each review are in blue, and some reviews are also shortened for space.

DATASET	$\lambda^t = 0$	$0 < \lambda^t < \infty$	$\lambda^t = \infty$
PATHOLOGY	84.1	94.1	77.0
REVIEW	80.9	86.4	84.3

Table 7: The effect of regularization of the transformation layer λ^t on the performance.

dicating that adversarial training helps learning of domain-invariant representations. However, adversarial training also removes a lot of information from representations, as the second matrix is much more sparse than the first one. The third matrix shows that adding reconstruction loss effectively addresses this sparsity issue. Almost 85% entries of the second matrix have small values ($< 10^{-6}$) while the sparsity is only about 30% for the third one. Moreover, the standard deviation of the third matrix is also ten times higher than the second one. These comparisons demonstrate that the reconstruction loss function improves both the richness and diversity of the learned representations. Note that in the case of no adversarial training (Ours-NA), adding the reconstruction component has no clear effect. This is expected because the main motivation of adding this component is to achieve a more robust adversarial training.

Regularization on the transformation layer Table 7 shows the averaged accuracy with different regularization weights λ^t in Equation 5. We change λ^t to reflect different model variants. First, $\lambda^t = \infty$ corresponds to the removal of the transformation layer because the transformation is always identity in this case. Our model performs better than this variant on both datasets, yielding an average improvement of 17.1% on the pathology dataset and 2.1% on the review dataset. This result indicates the importance of adding the transformation layer. Second, using zero regularization ($\lambda^t = 0$) also consistently results in inferior performance, such as 10% loss on the pathology dataset. We hypothesize that zero regularization will dilute the effect from reconstruction because of too much flexibility in transformation. As a result, the transformed representation will become sparse due to the adversarial training, leading to the performance loss.

Examples of neighboring reviews Finally, we illustrate in Figure 5 a case study on the characteristics of learned abstract representations by different models. The first column shows an example restaurant review. Sentiment phrases in this example are mostly food-specific, such as “undercooked” and “tasted weird”. In the other two columns, we show example hotel reviews that are nearest neighbors to the restaurant reviews, measured by cosine similarity between their representations. In column 2, many sentiment phrases are specific for room quality, such as “old” and “rained water from above”. In column 3, however, most sentiment phrases are either common sentiment expressions (e.g. dirty) or food-related (e.g. food poison), even though the focus of the reviews is room quality of hotels. This observation indicates that adversarial training (Ours-Full) successfully learns to eliminate domain-specific information and to map those domain-specific words into similar domain-invariant representations. In contrast, Ours-NA only captures domain-invariant features from phrases that commonly present in both domains.

7 Conclusions

In this paper, we propose a novel aspect-augmented adversarial network for cross-aspect and cross-domain adaptation tasks. Experimental results demonstrate that our approach successfully learns invariant representation from aspect-relevant fragments, yielding significant improvement over the mSDA baseline and our model variants. The effectiveness of our approach suggests the potential application of adversarial networks to a broader range of NLP tasks for improved representation learning, such as machine translation and language generation.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

John Blitzer, Mark Dredze, Fernando Pereira, et al. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Neural Information Processing Systems (NIPS)*.

Caroline Brun, Julien Perez, and Claude Roux. 2016. Xrce at semeval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 277–281.

Julliette M Buckley, Suzanne B Coopey, John Sharko, Fernanda Polubriaginof, Brian Drohan, Ahmet K Belli, Elizabeth MH Kim, Judy E Garber, Barbara L Smith, Michele A Gadd, et al. 2012. The feasibility of using natural language processing to extract clinical information from breast pathology reports. *Journal of pathology informatics*, 3(1):23.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *Annual Meeting-Association for Computational Linguistics*, volume 45, page 280.

Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*.

Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. 2015. Abc-cnn: An attention based convolutional neural network for visual question answering. *arXiv preprint arXiv:1511.05960*.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Sumit Chopra, Suhrid Balakrishnan, and Raghuraman Gopalan. 2013. Dlid: Deep learning for domain adaptation by interpolating between domains. In *ICML Workshop on Challenges in Representation Learning*.

Jacob Eisenstein. 2017. Unsupervised learning for lexicon-based classification. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2015. Domain-adversarial training of neural networks. *arXiv preprint arXiv:1505.07818*.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Trond Grenager, Dan Klein, and Christopher D Manning. 2005. Unsupervised learning of field segmentation models for information extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 371–378. Association for Computational Linguistics.

Aria Haghighi and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 320–327. Association for Computational Linguistics.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. 2012. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

Shen Li, Joao V Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods* *in Natural Language Processing and Computational Natural Language Learning*, pages 1389–1398. Association for Computational Linguistics.

Mingsheng Long, Jianmin Wang, and Michael I Jordan. 2016. Unsupervised domain adaptation with residual transfer networks. *arXiv preprint arXiv:1602.04433*.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Gideon S Mann and Andrew McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields.

Iain J Marshall, Joël Kuiper, and Byron C Wallace. 2015. Robotreviewer: evaluation of a system for automatically assessing bias in clinical trials. *Journal of the American Medical Informatics Association*, page ocv044.

André FT Martins and Ramón Fernandez Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. *arXiv preprint arXiv:1602.02068*.

Liron Pantanowitz, Maryanne Hornish, Robert A Goulart, et al. 2008. Informatics applied to cytology. *Cytojournal*, 5(1):16.

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. 2016. Learning transferrable representations for unsupervised domain adaptation. In *Advances In Neural Information Processing Systems*, pages 2110–2118.

Yusuke Shinohara. 2016. Adversarial multi-task learning of deep neural networks for robust speech recognition. *Interspeech 2016*, pages 2369–2372.

Jost Tobias Springenberg. 2015. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*.

Yaniv Taigman, Adam Polyak, and Lior Wolf. 2016. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 783–792. ACM.

Hongning Wang, Yue Lu, and ChengXiang Zhai. 2011. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 618–626. ACM.

Huijuan Xu and Kate Saenko. 2015. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. *arXiv preprint arXiv:1511.05234*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, page 5.

Adam Yala, Regina Barzilay, Laura Salama, Molly Griffin, Grace Sollender, Aditya Bardia, Constance Lehman, Julliette M Buckley, Suzanne B Coopey, Fernanda Polubriaginof, J Garber, BL Smith, MA Gadd, MC Specht, and TM Gudewicz. 2016. Using machine learning to parse breast pathology reports. *Breast Cancer Research and Treatment*.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2015. Stacked attention networks for image question answering. *arXiv preprint arXiv:1511.02274*.

Omar Zaidan, Jason Eisner, and Christine D Piatko. 2007. Using" annotator rationales" to improve machine learning for text categorization. In *HLT-NAACL*, pages 260–267. Citeseer.

Ye Zhang, Iain Marshall, and Byron C Wallace. 2016. Rationale-augmented convolutional neural networks for text classification. *arXiv preprint arXiv:1605.04469*.

Guangyou Zhou, Zhiwen Xie, Jimmy Xiangji Huang, and Tingting He. 2016. Bi-transferring deep neural networks for domain adaptation. ACL.