A Deep Learning Approach: Domain Adaptation for Large-Scale Sentiment Analysis

Meenakshi

Asst. Prof., Dept. of CSE, GGIAET, Gurgaon, Haryana

ABSTRACT

The Increase in the availability of online recommendations and reviews makes sentiment classification an interesting topic in industrial and academic research. So many different domains can be spanned by review and that is difficult to gather annotated training data for all of them. Hence, the problem of domain adaptation for sentiment classifiers is studied in this paper, Deep learning approach has been approached which learns to extract a meaningful representation for each review in an individual fashion.

I.INTRODUCTION

Surge of research in sentiment classification (or sentiment analysis) has been created with the rise of social media such as blogs and social networks, reviews, ratings and recommendations are rapidly proliferating; being able to automatically filter them is a current key challenge for businesses looking to sell their wares and identify new promote opportunities., It aims to determine the judgment of a writer with respect to a given topic based on a given textual comment. Now, Sentiment analysis is a mature machine learning research topic, as illustrated with this review (Pang and Lee, 2008). Many different domains have been presented by this application, ranging from movie reviews (Pang *et al.*, 2002) and congressional floor de- bates (Thomas *et al.*, 2006) to product recommendations (Snyder and Barzilay, 2007; Blitzer *et al.*, 2007).

Design a robust sentiment classifier through large variety of data sources makes it difficult and costly. Indeed, Vocabularies are different for reviews which deal with various kinds of products and services. For instance, considering the simple case of training a structure analyzing reviews about only two sorts of products: *home appliances* and *Movies*. One set of reviews would contain adjectives such as "failure", "guarantee" or "toughness", and the other "adventurous", "mind-blowing" or "disgusting", etc. Therefore, across domains, data distributions is different. One solution could be to learn a different system for each domain and this would imply a huge cost to annotate training data for a large number of domains and prevent us from exploiting the information shared across domains. An alternative solution, described here, consists in learning a single system from the set of domains for which labeled and unlabeled data are available and then apply it to any target domain (labeled or unlabeled). This only makes sense if the system is able to discover intermediate abstractions that are shared and meaningful across domains. This problem of testing and training models on different distributions is known as domain adaptation (Daumé III and Marcu, 2006).

We propose a Deep Learning approach for the problem of domain adaptation of sentiment classifiers in this paper. Recently, the promising new area of Deep Learning has been emerged; see (Bengio, 2009) for a review. It

is based on algorithms for **discovering intermediate representations** which is built in a Hierarchical manner. These features have successfully been used to initialize deep neural networks (Hinton and Salakhutdinov, 2006; Hinton *et al.*, 2006; Bengio *et al.*, 2006).

In Section 3 Deep Learning system is designed to use unlabeled data which is used extract high-level features from reviews. We show in Section 4 that sentiment classifiers trained with these learnt features can:

(i) Successfully perform domain adaptation on a large-scale data set of 22 domains, beating all of the baselines we tried and

(ii) Surpass state-of-the-art performance on a benchmark of 4 kinds of products.

II.DOMAIN REVISION

The testing and instruction data are sampled from a variety of distributions for which Domain adaptation considers the setting. Assume 2 sets of information: a source domain *A* providing labeled training instances and a *target* domain *B* given that instances on which the classifier is meant to be deployed. We do not make the hypothesis that these are drawn from the same distribution, but to a certain extent that *A* is drawn from a distribution p_a and *B* from a distribution p_b . The learning dilemma consists in finding a function realizing a good *convey* from *A* to *B* i.e. it is trained on data drawn from p_a and generalizes well on data drawn from p_b.

An transitional notion between unrefined input and target is being learned from Deep Learning algorithms. Our intuition for using it in this setting is that these intermediary concepts could acquiesce better transfer across domains. Suppose for example that these intermediate concepts indirectly capture things like product quality, product price, customer service, etc. Some of these concepts are general enough to make sense across a wide range it should be possible to discover them. Furthermore, because Deep Learning exploits unsupervised learning to find out these concepts, one can exploit the hefty amounts of unlabeled information across all domains to learn these intermediate representations. Here, as in many other Deep Learning approaches, we do not engineer what these transitional concepts should be, but in- stead use generic learning algorithms to determine them.

2.1 Literature Survey

Domain variation learning setups have been proposed before and published under different names. Daumé III and Marcu (2006) dignified the problem for which he proposed an approach based on a mixture model. Domain adaptation general way to address is through instance weighting, in which instance dependent weights are added to the loss function (Jiang and Zhai, 2007). Another solution to domain adaptation can be to transform the data representations of the source and target domains so that they present the same joint distribution of observations and labels. Ben-David *et al.* (2007) formally analyze the effect of representation change for domain adaptation while Blitzer *et al.* (2006) propose the Structural Correspondence Learning (SCL) algorithm that makes use of the unlabeled data from the target domain to find a low-rank joint representation of the data.

Finally, by ignoring the domain difference it can be simply treated as a standard semi-supervised problem and considering the source instances as labeled data and the target ones as unlabeled data (Dai *et al.*, 2007). In that case, the framework is very close to that of self-taught learning (Raina *et al.*, 2007), in which one learns from labeled examples of some categories as well as unlabeled examples from a larger set of categories. The approach of Raina *et al.* (2007) relies crucially on the unsupervised learning of a representation, like the approach proposed here.

2.2 Sentiment Classification Applications Survey

Sentiment analysis and domain adaptation are closely related in the literature, and many works have calculated domain adaptation completely for sentiment analysis. Among those, a huge majority propose experiments performed on the standard made of reviews of Amazon products gathered by Blitzer *et al.* (2007).

Amazon data The data set proposes more than 340,000 reviews regarding 22 different product types[1] and for which reviews are labeled as either positive of domains (corresponding to products or services, in the case of sentiment analysis). Because the same words or tuples of words may be used across domains to indicate the presence of these higher-level concepts.

Table 1. **Amazon data statistics.** This table depicts the number of training, testing and unlabeled examples for each domain, as well as the portion of negative training examples for both versions of the data set.

Domain	**Train size**	**Test size**	**Unlab. size**	**% Neg. ex**	**(Smaller-scale) benchmark**				
Complete (large-scale) data set					Books	1600	400	4465	50%
Toys	6318	2527	3791	19.63%	Kitchen	1600	400	5945	50%
Software	1032	413	620	37.77%	Electronics	1600	400	5681	50%
Apparel	4470	1788	2682	14.49%	DVDs	1600	400	3586	50%
Video	8694	3478	5217	13.63%					
Automotive	362	145	218	20.69%					
Books	10625	10857	32845	12.08%					
Jewelry	982	393	589	15.01%					
Grocery	1238	495	743	13.54%					
Camera	2652	1061	1591	16.31%					
Baby	2046	818	1227	21.39%					
Magazines	1195	478	717	22.59%					
Cell	464	186	279	37.10%					
Electronics	10196	4079	6118	21.94%					
DVDs	10625	9218	26245	14.16%					
Outdoor	729	292	437	20.55%					
Health	3254	1301	1952	21.21%					
Music	10625	24872	88865	8.33%					
Videogame	720	288	432	17.01%					

Kitchen	9233	3693	5540	20.96%
Beauty	1314	526	788	15.78%
Sports	2679	1072	1607	18.75%
Food	691	277	415	13.36%

There is a vast disparity between domains in the total number of instances and in the proportion of negative examples as detailed in Table 1(top)

Compared Methods In the original paper regarding the smaller 4-domain standard dataset, Blitzer *et al.* (2007) adapt Structural association Learning for sentiment analysis. Li and Zong (2008) propose the Multi-label Consensus Training (MCT) approach which combines several base classifiers trained with SCL. Pan *et al.* (2010) first use a Spectral Feature Alignment (SFA) algorithm to align words from different source and target domains to help bridge the gap between them. These 3 methods serve as comparisons in our empirical evaluation.

III.Deep Learning Approach

3.1 Conditions

To some extent, if deep learning algorithms are able to capture, the underlying generative factors that explain the variations in the input data, what is really needed to exploit that ability is for the learned representations to help in disentangling the underlying factors of variation. If some of the features learned (the individual elements of the learned representation) are mostly related to only some of these factors, perhaps only one then this could be the most useful and simplest way. Conversely, it would mean that such features would have *invariant properties*, i.e., they would be highly specific in their response to a sub- set (maybe only one) of these factors of variation and insensitive to the others. This hypothesis was tested by Goodfellow *et al.* (2009), for images and geometric invariance's associated with movements of the camera.

Evaluating Deep Learning algorithms on sentiment analysis is an interesting way for several reasons. First, if they can extract features that somewhat disentangle the underlying factors of variation, this would likely help to perform transfer across domains, since we expect that there exist generic concepts that characterize product reviews across many domains. Second, for our Amazon datasets, we know some of these factors (such as whether or not a review is about a particular product, or is a positive appraisal for that product), so we can use this knowledge to quantitatively check to what extent they are disentangled in the learned representation: domain adaptation for sentiment analysis becomes a medium for better understanding deep architectures. Finally, even though Deep Learning algorithms have not yet been evaluated for domain adaptation of sentiment classifiers, several very interesting results have been reported on other tasks involving textual data, beating the previous state-of-the-art in several cases (Salakhutdinov and Hinton, 2007; Collobert and Weston, 2008; Ranzato and Szummer, 2008).

3.2 Stacked Denoising Auto-encoders

Denoising autoencoders can be pushed to form a profound network by feeding the hidden illustration (output code) of the denoising autoencoder found on the layer under as input to the existing layer.SDA is an porch of the stacked autoencoder

Stacked Denoising Auto-encoder (Vincent *et al.*, 2008) is our essential framework model. An auto-encoder is comprised of an encoder function h(_) and a decoder function g(_), typically with the dimension of h(_) smaller than that of its argument. The reconstruction of input x is given by r(x) = g(h(x)), and auto-encoders are typically trained to minimize a form of reconstruction error loss(x; r(x)). Examples of reconstruction error include the squared error, or like here, when the elements of x or r(x) can be considered as probabilities of a discrete event, the Kullback Liebler divergence between elements of x and elements of r(x). When the encoder and decoder are linear and the reconstruction error is quadratic, one recovers in h(x) the space of the principal components (PCA) of x. Once an auto-encoder has been trained, one can stack another auto-encoder on top of it, by training a second one which sees the encoded output of the _rst one as its training data. Stacked auto-encoders were one of the

_rst methods for building deep architectures (Bengio et al., 2006), along with Restricted Boltzmann Machines (RBMs) (Hinton et al., 2006). Once a stack of auto-encoders or RBMs has been trained, their parameters describe multiple levels of representation for x and can be used to initialize a supervised deep neural network (Bengio, 2009) or directly feed a classier, as we do in this paper. An interesting alternative to the ordinary auto encoder is the Denoising Auto-encoder (Vincent et al., 2008) or DAE, in which the input vector x is stochastically corrupted into a vector ~x, and the model is trained to denoise, i.e., to minimize a denoising reconstruction error loss(x; r(~x)). Hence the DAE cannot simply copy its input ~x in its code layer h(~x), even if the dimension of h(~x) is greater than that of ~x. The denoising error can be linked in several ways to the likelihood of a generative model of the distribution of the uncorrupted examples x (Vincent, 2011).

3.3. Protocol

We have access to unlabeled data from various domains in our setting, and to the labels for one source domain only. With a two-step procedure we tackle the problem of domain adaptation for sentiment classifiers. First, a higher-level feature extraction is learnt in an unsupervised fashion from the text reviews of all the available domains using a Stacked Denoising Autoencoder (SDA) with rectifier units (i.e. max(0; x)) for the code layer. RBMs with (soft) rectifier units have been introduced in (Nair and Hinton, 2010). They have been shown to outperform other non-linearity's on a sentiment analysis task (Glorot et al., 2011) and for the same reason we have used such units. The SDA is learnt in a greedy layer-wise fashion using stochastic gradient descent. For the first layer, the non-linearity of the decoder is the logistic sigmoid, the corruption process is a masking noise (i.e. each active input has a probability P to be set to 0)2 and the training criterion is the Kullback-Liebler divergence. The rectifier nonlinearity is too hard to be used on \output" units: Reconstruction error gradients would not flow if the reconstruction was 0 when the target is positive. For training the DAEs of upper layers, we use the softplus activation function (i.e. log(1 + exp(x)), a smooth version of the rectifier) as non-linearity for the decoder output units. The squared error as reconstruction error criterion and a Gaussian corruption noise is also used by us, which is added before the rectifier non-linearity of the input layer in order to keep the sparsity of the representation. The code layer activations (after the rectifier), at different depths, define the new representations.

A linear classifier is trained on the transformed labeled data of the source domain in a second step. Support Vector Machines (SVM) being known to perform well on sentiment classification (Pang et al., 2002), we use a linear SVM with squared hinge loss. This classifier is eventually tested on the target domain(s).

IV. CONCLUSION

Demonstration has been given by this paper on a Deep Learning system based on Stacked Denoising Auto-Encoders. With sparse rectifier units which can perform an unsupervised feature extraction which is highly beneficial for the domain adaptation of sentiment classifiers.

REFERENCES

1. Ben-David, S., Blitzer, J., Crammer, K., and Sokolova, P. M. (2007). Analysis of representations for domain adaptation. In Advances in Neural Information Processing Systems 20 (NIPS'07).

2. Bengio, Y. (2009). Learning deep architectures for AI. Foundations and Trends in Machine Learning, 2(1), 1{127. Also published as a book. Now Publishers, 2009.

3. Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2006). Greedy layer-wise training of deep networks. In Adv. in Neural Inf. Proc. Syst. 19 , pages 153{160.

4.Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). heano: a CPU and GPU math expression compiler. In Proceedings of the Python for Scienti_c Computing Conference (SciPy). Oral.

5. Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'06).

6.Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classi_cation. In Proceedings of Association for Computational Linguistics (ACL'07).

7. Collobert, R. and Weston, J. (2008). A uni_ed architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of Internationnal Conference on Machine Learning 2008 , pages 160{167.

8. Dai, W., Xue, G.-R., Yang, Q., and Yu, Y. (2007). Transferring naive Bayes classi_ers for text classi_cation. In Proc. of Assoc. for the Adv. of Art. Int. (AAAI'07).

9. Daum_e III, H. and Marcu, D. (2006). Domain adaptation for statistical classi_ers. Journal of Arti_cial Intelligence Research, 26, 101{126.

10. Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse recti_er neural networks. In Proceeding of the Conference on Arti_cial Intelligence and Statistics.

11. Goodfellow, I., Le, Q., Saxe, A., and Ng, A. (2009). Measuring invariances in deep networks. In Advances in Neural Information Processing Systems 22 , pages 646{654.

12. Hinton, G. E. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. Science,313(5786), 504{507.

13. Hinton, G. E., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18, 1527{1554.

14. Jiang, J. and Zhai, C. (2007). Instance weighting for domain adaptation in nlp. In Proceedings of Association for Computational Linguistics (ACL'07).

15. Li, S. and Zong, C. (2008). Multi-domain adaptation for sentiment classi_cation: Using multiple classi_er combining methods. In Proc. of the Conference on Natural Language Processing and Knowledge Engineering.

16. Nair, V. and Hinton, G. E. (2010). Recti_ed linear units improve restricted boltzmann machines. In

Proceedings of the International Conference on Machine Learning.

17. Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., and Chen, Z. (2010). Cross-domain sentiment classi_cation via spectral feature alignment. In Proceedings of the International World Wide Web Conference (WWW'10).

18. Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2), 1{135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classi_cation using machine learningtechniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing.

19. Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y. (2007). Self-taught learning: transfer learning from unlabeled data. In proceedings of the International Conference on Machine Learning, pages 759{766.

20. Ranzato, M. and Szummer, M. (2008). Semi-supervised learning of compact document representations with deep networks. In Proceedings of the International Conference on Machine Learning (ICML'08), pages 792{799.