What's in a Domain?
Learning Domain-Robust Text Representations using Adversarial Training

Yitong Li and **Timothy Baldwin** and **Trevor Cohn**
School of Computing and Information Systems
The University of Melbourne, Australia
yitongl4@student.unimelb.edu.au, {tbaldwin,tcohn}@unimelb.edu.au

Abstract

Most real world language problems require learning from heterogenous corpora, raising the problem of learning robust models which generalise well to both similar (*in domain*) and dissimilar (*out of domain*) instances to those seen in training. This requires learning an underlying task, while not learning irrelevant signals and biases specific to individual domains. We propose a novel method to optimise both in- and out-of-domain accuracy based on joint learning of a structured neural model with domain-specific and domain-general components, coupled with adversarial training for domain. Evaluating on multi-domain language identification and multi-domain sentiment analysis, we show substantial improvements over standard domain adaptation techniques, and domain-adversarial training.

1 Introduction

Heterogeneity is pervasive in NLP, arising from corpora being constructed from different sources, featuring different topics, register, writing style, etc. An important, yet elusive, goal is to produce NLP tools that are capable of handling all types of texts, such that we can have, e.g., text classifiers that work well on texts from newswire to wikis to micro-blogs. A key roadblock is application to new domains, unseen in training. Accordingly, training needs to be robust to domain variation, such that domain-general concepts are learned in preference to domain-specific phenomena, which will not transfer well to out-of-domain evaluation. To illustrate, Bitvai and Cohn (2015) report learning formatting quirks of specific reviewers in a review text regression task, which are unlikely to prove useful on other texts.

This classic problem in NLP has been tackled under the guise of "domain adaptation", also known as unsupervised transfer learning, using feature-based methods to support knowledge transfer over multiple domains (Blitzer et al., 2007; Daumé III, 2007; Joshi et al., 2012; Williams, 2013; Kim et al., 2016). More recently, Ganin and Lempitsky (2015) proposed a method to encourage domain-general text representations, which transfer better to new domains.

Inspired by the above methods, in this paper we propose a novel technique for multitask learning of domain-general representations.[1] Specifically, we propose deep learning architectures for multi-domain learning, featuring a *shared* representation, and domain *private* representation. Our approach generalises the feature augmentation method of Daumé III (2007) to convolutional neural networks, as part of a larger deep learning architecture. Additionally, we use adversarial training such that the *shared* representation is explicitly discouraged from learning domain identifying information (Ganin and Lempitsky, 2015). We present two architectures which differ in whether domain is conditioned on or generated, and in terms of parameter sharing in forming private representations.

We primarily evaluate on the task of language identification ("LangID": Cavnar and Trenkle (1994)), using the corpora of Lui and Baldwin (2012), which combine large training sets over a diverse range of text domains. Domain adaptation is an important problem for this task (Lui and Baldwin, 2014; Jurgens et al., 2017), where text resources are collected from numerous sources, and exhibit a wide variety of language use. We show that while domain adversarial training overall improves over baselines, gains are modest. The same applies to twin shared/private architectures, but when the two methods are combined, we observe substantial improvements. Overall, our

[1] Code, data and evaluation scripts available at https://github.com/lrank/Domain_Robust_Text_Representation.git

Proceedings of NAACL-HLT 2018, pages 474–479
New Orleans, Louisiana, June 1 - 6, 2018. ©2018 Association for Computational Linguistics

methods outperform the state-of-the-art (Lui and Baldwin, 2012) in terms of out-of-domain accuracy. As a secondary evaluation, we use the Multi-Domain Sentiment Dataset (Blitzer et al., 2007), where we once again observe a clear advantage for our approaches, illustrating the potential of our technique more broadly in NLP.

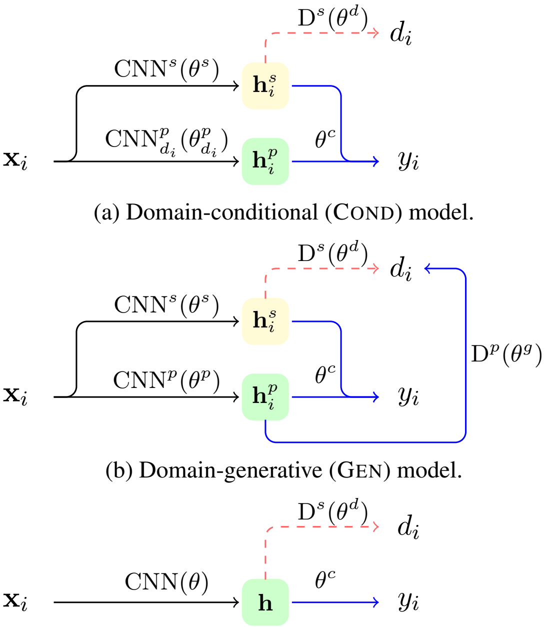
2 Multi-domain Learning

A primary consideration when formulating models of multi-domain data is how best to use the domain. Basic methods might learn several separate models, or simply ignore the domain and learn a single model. Neither method is ideal: the former fails to share statistics between the models to capture the general concept, while the latter discards information that can aid classification, e.g., domain-specific vocabulary or class skew.

To address these issues, we propose two architectures as illustrated in Figure 1 (a and b), parameterised as a convolutional network (CNN) over the input instance, chosen based on the success of CNNs for text categorisation problems (Kim, 2014); note, however, that our method is general and can be applied with other network types. Both representations are based on the idea of twin representations of each instance,[2] denoted *shared* and *private* representations, which are trained to capture domain-general versus domain-specific concepts, respectively. This is achieved using various loss functions, most notably an adversarial loss to discourage learning of domain-specific concepts in the shared representations. The two architectures differ in whether the domain is provided as an input (COND) or an output (GEN). Below, we elaborate on the details of the two models.

2.1 Domain-Conditional Model (COND)

The first model, illustrated in Figure 1a, includes a collection of domain-specific CNNs, and for each training instance $\mathbf{x}$, the domain-specific $\text{CNN}^p_{d_i}$ is used to compute its private representation $\mathbf{h}^p$. In this manner, the model *conditions* on the domain identifier. The COND model also computes a shared representation, $\mathbf{h}^s$, directly from $\mathbf{x}$, using a shared CNN^s, and the two representations are concatenated together to form input to linear softmax classification function c for predicting class label y. Thus far, the approach resembles Daumé III (2007), a method for multitask learning based on feature augmentation in a linear model, which works by replicating the input features to create both general shared features, and domain-specific features. Note that the approaches differ in that our method uses deep learning to form the two representations, in place of feature replication.



(a) Domain-conditional (COND) model.

(b) Domain-generative (GEN) model.

(c) Baseline CNN model with adversarial loss.

Figure 1: Proposed model architectures, showing a single training instance $(\mathbf{x}_i, y_i)$ with domain d_i, and baseline model with domain adversarial loss. CNN denotes a convolutional network, D indicates a discriminator (d for domain adversarial and g for domain generation), and red dashed and blue lines denote adversarial and standard loss, resp.

Adversarial Supervision A key challenge for the COND model is that the 'shared' representation can be contaminated by domain-specific concepts. To address this, we borrow ideas from adversarial learning (Goodfellow et al., 2014; Ganin et al., 2016). The central idea is to learn a good general representation (suitable for the shared component) to maximize end task performance, yet obscure the domain information, as modelled by a discriminator, D^s. This reduces the domain-specific information in the shared representation, however note that important domain-specific components can still be captured in the private representation.

Overall, this results in the training objective:

$$\mathcal{L}^{\text{COND}} = \min_{\theta^c,\theta^s,\{\theta^p\}} \max_{\theta^d} \mathcal{X}(\mathbf{y}|\mathbf{H}^s, \mathbf{H}^p, \mathbf{d}; \theta^c) - \lambda_d \mathcal{X}(\mathbf{d}|\mathbf{H}^s; \theta^d) \quad (1)$$

[2] This differs from standard architectures, e.g., 'baseline' in Figure 1c, which uses a single representation.

where $\mathcal{X}$ denotes the cross-entropy classification loss, $\mathbf{H}^s = \{\mathbf{h}_i^s(\mathbf{x}_i)\}_{i=1}^n$ are the shared representations for the training set of n instances, and likewise $\mathbf{H}^p = \{\mathbf{h}_i^p(\mathbf{x}_i, d_i)\}_{i=1}^n$ are the private representations, which are both functions of θ^s and $\{\theta_\cdot^p\}$, respectively. Note the negative sign of the adversarial loss (referred to as d), and the maximisation with respect to the discriminator parameters θ^d. This has the effect of learning a maximally accurate discriminator *wrt* θ^d, while making it maximally inaccurate *wrt* representation $\mathbf{H}^s$, and is implemented using a gradient reversal step during backpropagation (Ganin et al., 2016).

Minimum Entropy Inference As COND conditions on the domain, this imposes the requirement that the domain of the test data is known (and covered in training), which is incompatible with our goal of unsupervised adaptation. To deal with this situation, we consider each domain in the test set as belonging to one of the training domains, and then select the domain with the minimum entropy classification distribution. This is based on an assumption that a closely matching domain should be able to make confident predictions.[3]

2.2 Domain-Generative Model (GEN)

The second model is based on *generation* of, rather than *conditioning* on, the domain, which allows the model to learn domain signals that transfer across some, but not all, domains. Most components are common with the COND model as described in §2.1, including the use of private and shared representations, their use in the classification output, and the adversarial loss based on discriminating the domain from the shared representation. There are two key differences: (1) the private representation, $\mathbf{h}^p$, is computed using a single CNN^p, rather than several domain-specific CNNs, which confers benefits of domain-generalisation, a more compact model, and simpler test inference;[4] and (2) the private representation is used to positively predict the domain, which further encourages the split between domain general and domain-specific aspects of the representation.

GEN has the following training objective,

$$\begin{aligned}\mathcal{L}^{\text{GEN}} = \min_{\theta^c,\theta^s,\theta^p,\theta^g} \max_{\theta^d} &\mathcal{X}(\mathbf{y}|\mathbf{H}^s, \mathbf{H}^p; \theta^c) \\ &- \lambda_d \mathcal{X}(\mathbf{d}|\mathbf{H}^s; \theta^d) + \lambda_g \mathcal{X}(\mathbf{d}|\mathbf{H}^p; \theta^g)\end{aligned} \tag{2}$$

where notation follows that used in §2.1, with the exception of $\mathbf{H}^p = \{\mathbf{h}_i^p(\mathbf{x}_i)\}_{i=1}^n$ that is redefined, with $\mathbf{h}_i^p(\mathbf{x}_i)$ a function of θ^p, and the addition of the last term to capture the generation loss g. The same gradient reversal method from §2.1 is used during training for the adversarial component.

3 Experiments

3.1 Language Identification

To evaluate our approach, we first consider the language identification task.

Data We follow the settings of Lui and Baldwin (2012), involving 5 training sets from 5 different domains with 97 languages in total: Debian, JRC-Acquis, Wikipedia, ClueWeb and RCV2, derived from Lui and Baldwin (2011).[5] We evaluate accuracy on seven holdout benchmarks: EuroGov, TCL, Wikipedia2[6] (all from Baldwin and Lui (2010)), EMEA (Tiedemann, 2009), EuroPARL (Koehn, 2005), T-BE (Tromp and Pechenizkiy, 2011), and T-SC (Carter et al., 2013).

Documents are tokenized as a byte sequence (consistent with Lui and Baldwin (2012)), and truncated or padded to a length of 1k bytes.[7]

Hyper-parameters We perform a grid search for the hyper-parameters, and selected the following settings to optimise accuracy over heldout data from each of the training domains. All byte tokens are mapped to byte embeddings, which are random initialized with size 300. We use the filter sizes of $2, 4, 8, 16$ and 32, with 128 filters for each, to capture n-gram features of different lengths. A dropout rate of 0.5 was applied to all the representation layers. We set the factors λ_d and λ_g to 10^{-3}. All the models are optimized using the Adam Optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-4}.

[3] The minimum entropy method is quite effective, trailing oracle selection by only 0.8% accuracy.

[4] The domain need not be known for test examples, so the model can be used directly.

[5] As ClueWeb in Lui and Baldwin (2012) is not publicly accessible, we used a slightly different set of languages but comparable number of documents for training.

[6] Note that the two Wikipedia datasets have no overlap.

[7] We also tried different document length limits, such as 10k, but observed no substantial change in performance.

Models	EuroGov	TCL	Wikipedia2	EMEA	EuroPARL	T-BE	T-SC	ALL$_{\text{out}}$
baseline CNN	**99.9**	91.7	88.9	93.1	98.2	85.2	92.2	92.7
$+d$	**99.9**	92.4	88.4	90.2	98.2	87.7	93.1	92.8
$+g$	**99.9**	92.0	88.7	91.6	98.4	86.8	92.8	92.9
Cond	**99.9**	91.3	88.2	92.0	98.7	91.5	94.5	93.7
$+d$	**99.9**	**93.5**	90.1	91.3	98.7	92.6	**97.9**	**94.9**
Gen	**99.9**	92.3	88.0	93.3	98.6	87.1	93.8	93.3
$+d+g$	**99.9**	93.1	88.7	92.5	99.1	91.2	96.1	94.4
langid.py	98.7	90.4	**91.3**	**93.4**	**99.2**	**94.1**	88.6	93.6
cld2	99.0	85.0	85.3	90.7	98.5	85.0	93.4	91.0

Table 1: Accuracy [%] of the different models over the seven heldout datasets, and the macro-averaged accuracy out-of-domain over the 7 test domains ("ALL$_{\text{out}}$"). The best result for each dataset is indicated in **bold**. Key: $+d$ = domain adversarial, $+g$ = domain generation component.

3.1.1 Results and Analysis

Baseline and comparisons For comparison, we implement a CNN baseline[8] which is trained using all the data without domain knowledge (i.e. the simple union of the different training datasets). We also employ adversarial learning (d) and generation (g) of domain to the baseline model to better understand the utility of these methods. Note that the baseline +d is a multi-domain variant of Ganin and Lempitsky (2015), albeit trained without any text in the testing domains. For our models, we report results of configurations both with and without the d and g components. We also report the results for two state-of-the-art off-the-shelf LangID tools: (1) langid.py[9] (Lui and Baldwin, 2012); and (2) Google's cld2.[10]

Out-of-domain Results Our primary concern in terms of evaluating the ability of the different models to generalise, is out-of-domain performance. Table 1 provides a breakdown of out-of-domain results over the 7 holdout domains. The accuracy varies greatly between test domains, depending on the mix of languages, length of test documents, etc. Both our models, Cond and Gen, achieve competitive performance, and are further improved by d and g.

For the baseline, applying either d or g results in mild improvements over the baseline, which is surprising as the two forms of supervision work in opposite directions. Overall the small change in performance means neither method appears to be a viable technique for domain adaptation.

Overall, the raw Cond and Gen perform better than the baseline. Specifically, for Cond, we observed performance gains on EuroPARL, T-BE and T-SC. These three datasets are notable in containing shorter documents, which benefit the most from shared learning. However, as discussed earlier, multi-domain data can introduce noise to the shared representation, causing the performance to drop over TCL, Wikipedia2 and EMEA. This observation demonstrates the necessity of applying adversarial learning to Cond. On the other hand, it is a different story for Gen: vanilla Gen achieves accuracy gains relative to the baseline over 5 domains, but is slightly below Cond for 4 domains, a result of parameter-sharing over the private representation.

In terms of the adversarial learning, we see that by adding an adversarial component ($+d$ or $+d+g$), Cond and Gen realises substantial improvements out of domain, with the exception of EMEA. As we motivated, the domain adversarial part d can obscure the domain-specific information in the shared representation, which helps Cond have better generalisation to other domains. Additionally, applying g to Gen helps the private representation to generalize better. These results demonstrate that both d and g are necessary components of multi-domain models. EMEA is noteworthy in that its pattern of results is overall different to the other domains, in that applying d hurts performance. For this domain, the baseline CNN performs very well, and Gen does much better than Cond. We believe the reason is that, as a medical domain, EMEA is very much an outlier and does not align to any single training domain. Also, there is a lot of borrowing of terms such as drug and disease names verbatim between lan-

[8] The baseline here used a double capacity hidden representation, in order to better match the increased expressivity of the shared/private models.

[9] https://github.com/saffsd/langid.py

[10] https://github.com/CLD2Owners/cld2

Models	Deb	JRCA	Wiki	ClWb	RCV2	ALL_{in}
baseline CNN	96.6	99.8	97.8	90.7	97.9	96.6
baseline $+d$	96.5	99.8	97.8	90.7	97.0	96.4
COND	97.0	99.9	97.8	90.9	98.0	96.7
COND $+d$	97.0	**99.9**	**98.4**	90.8	98.1	96.8
GEN $+d+g$	**97.8**	**99.9**	98.0	91.1	97.9	96.9
LANGID.PY	97.4	99.8	97.6	**91.3**	**99.3**	**97.1**
CLD2	92.2	99.8	92.3	92.3	89.8	93.3

Table 2: Accuracy [%] of different models over five in-domain datasets using cross-validation evaluation and macro-averaged accuracy ("ALL_{in}").

guages, further complicating the task.

Overall, our best models (COND $+d$ and GEN $+d+g$) outperform both LANGID.PY and CLD2 in terms of average out-of-domain accuracy.

In-domain Results Table 2 reports the in-domain performance over the 5 training domains, using 5-fold cross validation, as well as the macro-averaged accuracy. Our proposed methods (COND $+d$ and COND $+d+g$) consistently achieve better performance than the baseline. Both COND and GEN achieve competitive performance with the state-of-the-art LANGID.PY in the in-domain scenario. Although LANGID.PY performs slightly better on average accuracy, our best model outperforms LANGID.PY for three of the five datasets.

3.2 Product Reviews

To evaluate the generalization of our methods to other tasks, we experiment with the Multi-Domain Sentiment Dataset (Blitzer et al., 2007).[11] We select the 20 domains with the most review instances, and discard the remaining 5 domains.

For model parameterization, we adopt the same basic hyper-parameter settings and training process as for LangID in §3.1, but change the filter sizes to 3, 4 and 5, use word-based tokenisation, and truncate sentences to 256 tokens, for better compatible with shorter documents.

We perform a out-of-domain evaluation over four target domains, "book" (B), "dvd" (D), "electronics" (E) and "kitchen & housewares" (K), as used in Blitzer et al. (2007). Our experimental setup differs from theirs, in that they train on a single domain and then evaluate on another, while we train over 16 domains, then evaluate on the four test domains.

Models	B	D	E	K
baseline CNN	79.6	81.2	86.3	87.2
baseline $+d$	78.7	81.6	86.6	87.1
COND	79.2	81.8	85.8	87.2
COND $+d$	79.8	82.3	86.8	87.4
GEN $+d+g$	**80.2**	**82.4**	**87.3**	**87.8**
SCL_MI ♣	76.0	78.5	77.9	85.9
DANN ◇	72.3	78.4	84.3	85.4
IN DOMAIN ♣	82.4	80.4	84.4	87.7

Table 3: Accuracy [%] of different models over 4 domains (B, D, E and K) under out-of-domain evaluations on Multi Domain Sentiment Dataset. Key: ♣ from Blitzer et al. (2007); ◇ from Ganin and Lempitsky (2015).

Table 3 presents the results. Overall, our proposed methods consistently outperform the baselines, with the GEN $+d+g$ approach a consistent winner over all other techniques. Note also the lacklustre performance when the baseline is trained with the adversarial loss, mirroring our findings for language identification in §3.1. For comparison, we also report the best results of SCL-MI and DANN, in both cases using an oracle selection of source domain. Our method consistently outperform these approaches, despite having no test oracle, although note that we use more diverse data sources for training.

4 Conclusions

We have proposed a novel deep learning method for multi-domain learning, based on joint learning of domain-specific and domain-general components, using either domain conditioning or domain generation. Based on our evaluation over multi-domain language identification and multi-domain sentiment analysis, we show our models to substantially outperform a baseline deep learning method, and set a new benchmark for state-of-the-art cross-domain LangID. Our approach has potential to benefit other NLP applications involving multi-domain data.

Acknowledgments

We thank the anonymous reviewers for their helpful feedback and suggestions, and the National Computational Infrastructure Australia for computation resources. This work was supported by the Australian Research Council (FT130101105).

[11] From `https://www.cs.jhu.edu/~mdredze/datasets/sentiment/`, using the positive and negative files from `unprocessed`, up to 2,000 instances per domain. For the four test domains we automatically aligned the reviews in the `processed` and `unprocessed`, such that we can compare results directly against prior work.

References

Timothy Baldwin and Marco Lui. 2010. Language identification: The long and the short of the matter. In *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*. pages 229–237.

Zsolt Bitvai and Trevor Cohn. 2015. Non-linear text regression with a deep convolutional neural network. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing Short Papers*.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. pages 440–447.

Simon Carter, Wouter Weerkamp, and Manos Tsagkias. 2013. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation* 47(1):195–215.

William B Cavnar and John M Trenkle. 1994. N-gram-based text categorization. In *Proceedings of the Third Symposium on Document Analysis and Information Retrieval*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. pages 256–263.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning 2015*. pages 1180–1189.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17:59:1–59:35.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*. pages 2672–2680.

Mahesh Joshi, Mark Dredze, William W. Cohen, and Carolyn Penstein Rosé. 2012. Multi-domain learning: When do domains matter? In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. pages 1302–1312.

David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017. Incorporating dialectal variability for socially equitable language identification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. pages 51–57.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. pages 1746–1751.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2016. Frustratingly easy neural domain adaptation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. pages 387–396.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit 2005*. pages 79–86.

Marco Lui and Timothy Baldwin. 2011. Cross-domain feature selection for language identification. In *Fifth International Joint Conference on Natural Language Processing*. pages 553–561.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of ACL 2012 System Demonstrations*. pages 25–30.

Marco Lui and Timothy Baldwin. 2014. Accurate language identification of Twitter messages. In *Proceedings of the 5th workshop on language analysis for social media*. pages 17–25.

Jörg Tiedemann. 2009. News from OPUS – a collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*. volume 5, pages 237–248.

Erik Tromp and Mykola Pechenizkiy. 2011. Graph-based n-gram language identification on short texts. In *Proceedings of the 20th Machine Learning Conference of Belgium and The Netherlands*. pages 27–34.

Jason Williams. 2013. Multi-domain learning and generalization in dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*. pages 433–441.