Unsupervised Domain Adaptation for Neural Machine Translation with Domain-Aware Feature Embeddings

Zi-Yi Dou, Junjie Hu, Antonios Anastasopoulos, Graham Neubig
Language Technologies Institute, Carnegie Mellon University
{zdou, junjieh, aanastas, gneubig}@cs.cmu.edu

Abstract

The recent success of neural machine translation models relies on the availability of high quality, in-domain data. Domain adaptation is required when domain-specific data is scarce or nonexistent. Previous unsupervised domain adaptation strategies include training the model with in-domain copied monolingual or back-translated data. However, these methods use generic representations for text regardless of domain shift, which makes it infeasible for translation models to control outputs conditional on a specific domain. In this work, we propose an approach that adapts models with domain-aware feature embeddings, which are learned via an auxiliary language modeling task. Our approach allows the model to assign domain-specific representations to words and output sentences in the desired domain. Our empirical results demonstrate the effectiveness of the proposed strategy, achieving consistent improvements in multiple experimental settings. In addition, we show that combining our method with back translation can further improve the performance of the model.[1]

1 Introduction

While neural machine translation (NMT) systems have proven to be effective in scenarios where large amounts of in-domain data are available (Gehring et al., 2017; Vaswani et al., 2017; Chen et al., 2018), they have been demonstrated to perform poorly when the test domain does not match the training data (Koehn and Knowles, 2017). Collecting large amounts of parallel data in all possible domains we are interested in is costly, and in many cases impossible. Therefore, it is essential to explore effective methods to train models that generalize well to new domains.

[1]Our code is publicly available at: https://github.com/zdou0830/DAFE.

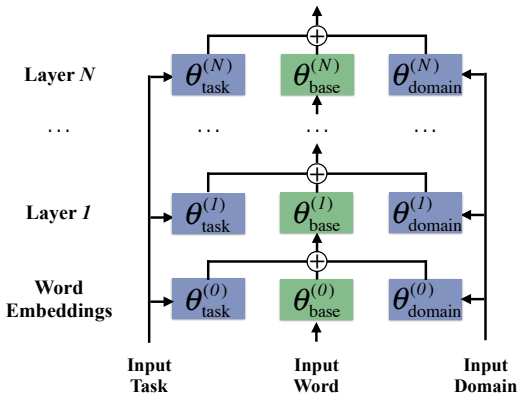

Figure 1: Main architecture of DAFE. Embedding learners generate domain- and task-specific features at each layer, which are then integrated into the output of the base network.

Domain adaptation for neural machine translation has attracted much attention in the research community, with the majority of work focusing on the supervised setting where a small amount of in-domain data is available (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016; Chu et al., 2017; Vilar, 2018). An established approach is to use domain tags as additional input, with the domain representations learned over parallel data (Kobus et al., 2017). In this work, we focus on *unsupervised* adaptation, where there are no in-domain parallel data available. Within this paradigm, Currey et al. (2017) copy the in-domain monolingual data from the target side to the source side and Sennrich et al. (2016a) concatenate back-translated data with the original corpus. However, these methods learn generic representations for all the text, as the learned representations are shared for all the domains and synthetic and natural data are treated equally. Sharing embeddings may be sub-optimal as data from different domains are inherently different. This problem is exacerbated

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1417–1422,
Hong Kong, China, November 3–7, 2019.

when words have different senses in different domains.

In this work, we propose a method of *Domain-Aware Feature Embedding* (DAFE) that performs unsupervised domain adaptation by disentangling representations into different parts. Because we have no in-domain parallel data, we learn DAFEs via an auxiliary task, namely language modeling. Specifically, our proposed model consists of a base network, whose parameters are shared across settings, as well as both domain and task embedding learners. By separating the model into different components, DAFE can learn representations tailored to specific domains and tasks, which can then be utilized for domain adaptation.

We evaluate our method in a Transformer-based NMT system (Vaswani et al., 2017) under two different data settings. Our approach demonstrates consistent improvements of up to 5 BLEU points over unadapted baselines, and up to 2 BLEU points over strong back-translation models. Combining our method with back translation can further improve the performance of the model, suggesting the orthogonality of the proposed approach and methods that rely on synthesized data.

2 Methods

In this section, we first illustrate the architecture of DAFE, then describe the overall training strategy.

2.1 Architecture

DAFE disentangles hidden states into different parts so that the network can learn representations for particular domains or tasks, as illustrated in Figure 1. Specifically, it consists of three parts: a *base network* with parameters θ_{base} that learns common features across different tasks and domains, a *domain-aware feature embedding learner* that generates embeddings $\theta^{\tau}_{\text{domain}}$ given input domain τ and a *task-aware feature embedding learner* that outputs task representations $\theta^{\gamma}_{\text{task}}$ given input task γ. The final outputs for each layer are obtained by a combination of the base network outputs and feature embeddings.

The base network is implemented in the encoder-decoder framework (Sutskever et al., 2014; Cho et al., 2014). Both the task and domain embedding learners directly output feature embeddings with look-up operations.

In this work, the domain-aware embedding learner learns domain representations $\theta^{in}_{\text{domain}}$ and $\theta^{out}_{\text{domain}}$ from in-domain and out-of-domain data respectively, and the task-aware embedding leaner learns task embeddings $\theta^{mt}_{\text{task}}$ and $\theta^{lm}_{\text{task}}$ for machine translation and language modeling.

Algorithm 1 Training Strategy

while $\theta_{\text{base}}, \theta_{\text{domain}}, \theta_{\text{task}}$ have not converged **do**
  Sample $\{(C(\mathbf{y}), \mathbf{y})\}$ from Y^{in}
  Train $\{\theta_{\text{base}}, \theta^{in}_{\text{domain}}, \theta^{lm}_{\text{task}}\}$ with Eqn. 2
  Sample $\{(C(\mathbf{y}), \mathbf{y})\}$ from Y^{out}
  Train $\{\theta_{\text{base}}, \theta^{out}_{\text{domain}}, \theta^{lm}_{\text{task}}\}$ with Eqn. 2
  Sample $\{(\mathbf{x}, \mathbf{y})\}$ from (X^{out}, Y^{out})
  Train $\{\theta_{\text{base}}, \theta^{out}_{\text{domain}}, \theta^{mt}_{\text{task}}\}$ with Eqn. 1
end while

The feature embeddings are generated at *each* encoding layer (including the source word embedding layer) and have the same size as the hidden states of the base model. It should be noted that feature embedding learners generate different embeddings at different layers.

Formally, given a specific domain τ and task γ, the output of the l-th encoding layer $\mathbf{H}^{(l)}_e$ would be:

$$\mathbf{H}^{(l)}_e = \textsc{Layer}_e(\mathbf{H}^{(l-1)}_e; \theta^{(l)}_{\text{base}}) + \theta^{\gamma,(l)}_{\text{domain}} + \theta^{\tau,(l)}_{\text{task}},$$

where $\theta^{\gamma,(l)}_{\text{domain}}$ and $\theta^{\tau,(l)}_{\text{task}}$ are single vectors and $\textsc{Layer}_e(\cdot)$ can be any layer encoding function, such as an LSTM (Hochreiter and Schmidhuber, 1997) or Transformer (Vaswani et al., 2017).

In this paper, we adopt a simple, add operation to combine outputs of different parts which already achieves satisfactory performance as shown in Section 3. We leave investigating more sophisticated combination strategies for future work.

2.2 Training Objectives

In the unsupervised domain adaptation setting, we assume access to an out-of-domain parallel training corpus (X^{out}, Y^{out}) and target-language in-domain monolingual data Y^{in}.

Neural machine translation. Our target task is machine translation. Both the base network and embedding learners are jointly trained with the objective:

$$\max_{\theta} \sum_{(\mathbf{x},\mathbf{y})\in(X^{out},Y^{out})} \log p(\mathbf{y}|\mathbf{x};\theta), \qquad (1)$$

where $\theta = \{\theta_{\text{base}}, \theta^{mt}_{\text{task}}, \theta^{out}_{\text{domain}}\}$.

Method	German-English						Czech-English			German-English		
	LAW		MED		IT		WMT			WMT		
	MED	IT	LAW	IT	LAW	MED	TED	LAW	MED	TED	LAW	MED
(1) Baseline	13.25	5.22	4.55	3.15	4.29	7.05	24.30	28.22	15.45	28.15	24.61	26.75
Data-Centric Methods												
(2) Copy	19.23	7.57	6.01	5.89	5.11	11.15	26.44	36.49	22.73	29.79	26.17	29.33
(3) Back	22.53	11.34	7.62	7.02	8.06	14.56	32.70	42.08	30.45	34.46	30.24	33.16
Model-Centric Methods												
(4) DAFE w/o Embed	23.44	6.97	9.21	8.34	8.09	16.34	26.63	35.86	23.44	29.88	27.10	32.15
(5) DAFE	24.23	8.59	9.87	8.44	8.61	17.50	28.09	38.89	26.05	30.88	27.77	32.48
Combining Data-Centric and Model-Centric Methods												
(6) Back + DAFE	25.34	13.55	9.60	11.20	9.60	17.25	**33.18**	44.06	34.24	34.57	30.72	35.48
(7) Back-DAFE	26.47	13.75	11.90	9.47	10.60	18.04	32.51	43.33	35.45	34.57	30.93	37.66
(8) Back-DAFE + DAFE	**26.96**	**15.41**	**14.28**	**13.03**	**11.67**	**21.30**	33.02	**44.36**	**37.48**	**34.89**	**31.46**	**38.79**

Table 1: Translation accuracy (BLEU) under different settings. The second and third rows list source and target domains respectively. "DAFE w/o Embed" denotes DAFE without embedding learners and "Back-DAFE" denotes back-translation by target-to-source model trained with DAFE. DAFE outperforms other approaches when adapting between domains (row 1-5, column 2-7) and is complementary to back-translation (row 6-8).

Language modeling. We choose masked language modeling (LM) as our auxiliary task. Following Lample et al. (2018a,b), we create corrupted versions $C(\mathbf{y})$ for each target sentence $\mathbf{y}$ by randomly dropping and slightly shuffling words. During training, gradient ascent is used to maximize the objective:

$$\max_{\theta} \sum_{\mathbf{y} \in \{Y^{in} \cup Y^{out}\}} \log p(\mathbf{y}|C(\mathbf{y}); \theta), \quad (2)$$

where $\theta = \{\theta_{\text{base}}, \theta^{lm}_{\text{task}}, \theta^{out}_{\text{domain}}\}$ for out-of-domain data and $\{\theta_{\text{base}}, \theta^{lm}_{\text{task}}, \theta^{in}_{\text{domain}}\}$ for in-domain data.

Training strategy. Our training strategy is shown in Algorithm 1. The ultimate goal is to learn a set of parameters $\{\theta_{\text{base}}, \theta^{in}_{\text{domain}}, \theta^{mt}_{\text{task}}\}$ for in-domain machine translation. While out-of-domain parallel data allows us to train $\{\theta_{\text{base}}, \theta^{out}_{\text{domain}}, \theta^{mt}_{\text{task}}\}$, the monolingual data help the model learn both $\theta^{in}_{\text{domain}}$ and $\theta^{out}_{\text{domain}}$.

3 Experiments

3.1 Setup

Datasets. We validate our models in two different data settings. First, we train on the law, medical and IT datasets of the German-English OPUS corpus (Tiedemann, 2012) and test our methods' ability to adapt from one domain to another. The dataset contain 2K development and test sentences in each domain, and about 715K, 1M and 337K training sentences respectively. These datasets are relatively small and the domains are quite distant from each other. In the second setting, we adapt models trained on the general-domain WMT-14 datasets into both the TED (Duh, 2018) and law, medical OPUS datasets. For this setting, we consider two language pairs, namely Czech and German to English. The Czech-English and German-English datasets consist of 1M and 4.5M sentences and the development and test sets contain about 2K sentences.

Models. We implement DAFE on top of the Transformer model. Both the encoder and decoder consist of 4 layers and the hidden size is set to 512. Byte-pair encoding (Sennrich et al., 2016b) is employed to process training data into subwords for a final shared vocabulary size of 50K.

Baselines. We compare our methods with two baseline models: 1) The copied monolingual data model (Currey et al., 2017) which copies target in-domain monolingual data to the source side; 2) Back-translation (Sennrich et al., 2016a) which enriches the training data by generating synthetic in-domain parallel data via a target-to-source NMT model. We characterize the two baselines as *data-centric* methods as they rely on synthesized data. In contrast, our approach is *model-centric* as we mainly focus on modifying the model architecture. We also perform an ablation study by removing the embedding learners (denoted as "DAFE w/o Embed") and the model will just perform multi-task learning.

3.2 Main Results

Adapting between domains. As shown in the first 6 results columns of Table 1, the unadapted baseline model (row 1) performs poorly when adapting between domains. The copy method

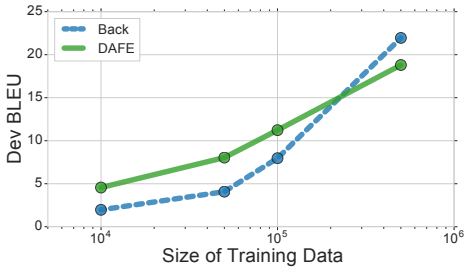

Figure 2: DAFE outperforms back-translation in low-resource scenarios.

(row 2) and back translation (row 3) both improve the model significantly, with back-translation being a better alternative to copying. DAFE (row 5) achieves superior performance compared to back-translation, with improvements of up to 2 BLEU points. Also, removing the embedding learners leads to degraded performance (row 4), indicating the necessity of their existence.

Adapting from a general to a specific domain. In the second data setting (last 6 columns of Table 1), with relatively large amounts of general-domain datasets, back-translation achieves competitive performance. In this setting, DAFE improves the unadapted baseline significantly but it does not outperform back-translation. We hypothesize this is because the quality of back-translated data is relatively good.

3.3 Combining DAFE with Back-Translation

We conjecture that DAFE is complementary to the data-centric methods. We attempt to support this intuition by combining DAFE with back-translation (the best data-centric approach). We try three different strategies to combine DAFE with back-translation, outlined in Table 1.

Simply using back-translated data to train DAFE (row 6) already achieves notable improvements of up to 4 BLEU points. We can also generate back-translated data using target-to-source models trained with DAFE, with which we train the forward model (Back-DAFE, row 7). By doing so, the back-translated data will be of higher quality and thus the performance of the source-to-target model can be improved. The overall best strategy is to use Back-DAFE to generate synthetic in-domain data and train the DAFE model with the back-translated data (Back-DAFE+DAFE, row 8). Across almost all adaptation settings, Back-DAFE+DAFE leads to higher translation quality, as per our intuition. An advantage of this setting is that the back-translated data allow us to learn $\theta^{in}_{\text{domain}}$ with the translation task.

Embedding	MED dev	MED test	IT dev	IT test
MED	42.06	34.63	4.13	4.80
IT	36.96	30.09	7.54	8.44

Table 2: Providing mismatched domain embeddings leads to degraded performance.

Reference	please report this bug to the developers .
MED-embed	please report this to the EMEA .
IT-embed	please report this bug to the developers .
Reference	for intramuscular use .
MED-embed	for intramuscular use .
IT-embed	for the use of the product .

Table 3: Controlling the output domain by providing different domain embeddings. We use compare-mt (Neubig et al., 2019) to select examples.

3.4 Analysis

Low-resource scenarios. One advantage of DAFE over back-translation is that we do not need a good target-to-source translation model, which can be difficult be acquire in low-resource scenarios. We randomly sample different amounts of training data and evaluate the performance of DAFE and back-translation on the development set. As shown in Figure 2, DAFE significantly outperforms back-translation in data-scarce scenarios, as low quality back-translated data can actually be harmful to downstream performance.

Controlling the output domain. An added perk of our model is the ability to control the output domain by providing the desired domain embeddings. As shown in Table 2, feeding mismatched domain embeddings leads to worse performance. Examples in Table 3 further suggest the model with medical embeddings as input can generate domain-specific words like "EMEA" (European Medicines Evaluation Agency) and "intramuscular", while IT embeddings encourage the model to generate words like "bug" and "developers".

4 Related Work

Most previous domain adaptation work for NMT focus on the setting where a small amount of in-domain data is available. Continued training (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016) methods first train an NMT model on out-of-domain data and then fine-tune it on the in-domain data. Similar to our work, Kobus et al.

(2017) propose to use domain tags to control the output domain, but it still needs a in-domain parallel corpus and our architecture allows more flexible modifications than just adding additional tags.

Unsupervised domain adaptation techniques for NMT can be divided into data- and model-centric methods (Chu and Wang, 2018). Data-centric approaches mainly focus on selecting or generating the domain-related data using existing in-domain monolingual data. Both the copy method (Currey et al., 2017) and back-translation (Sennrich et al., 2016a) are representative data-centric methods. In addition, Moore and Lewis (2010); Axelrod et al. (2011); Duh et al. (2013) use LMs to score the out-of-domain data, based on which they select data similar to in-domain text. Model-centric methods have not been fully investigated yet. Gulcehre et al. (2015) propose to fuse LMs and NMT models, but their methods require querying two models during inference and have been demonstrated to underperform the data-centric ones (Chu et al., 2018). There are also work on adaptation via retrieving sentences or n-grams in the training data similar to the test set (Farajian et al., 2017; Bapna and Firat, 2019). However, it can be difficult to find similar parallel sentences in domain adaptation settings.

5 Conclusion

In this work, we propose a simple yet effective unsupervised domain adaptation technique for neural machine translation, which adapts the model by domain-aware feature embeddings learned with language modeling. Experimental results demonstrate the effectiveness of the proposed approach across settings. In addition, analysis reveals that our method allows us to control the output domain of translation results. Future work include designing more sophisticated architectures and combination strategies as well as validating our model on other language pairs and datasets.

Acknowledgements

We are grateful to Xinyi Wang and anonymous reviewers for their helpful suggestions and insightful comments. We also thank Zhi-Hao Zhou, Shuyan Zhou and Anna Belova for proofreading the paper.

This material is based upon work generously supported partly by the National Science Foundation under grant 1761548 and the Defense Advanced Research Projects Agency Information Innovation Office (I2O) Low Resource Languages for Emergent Incidents (LORELEI) program under Contract No. HR0011-15-C0114. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ankur Bapna and Orhan Firat. 2019. Non-parametric adaptation for neural machine translation. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2018. A comprehensive empirical comparison of domain adaptation methods for neural machine translation. *Journal of Information Processing*, 26:529–538.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *International Conference on Computational Linguistics (COLING)*.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Conference on Machine Translation (WMT)*.

Kevin Duh. 2018. The multitarget ted talks task. http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

M Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Conference on Machine Translation (WMT)*.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning (ICML)*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *International Conference Recent Advances in Natural Language Processing (RANLP)*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Workshop on Neural Machine Translation (WMT)*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, et al. 2018b. Phrase-based & neural unsupervised machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *International Workshop on Spoken Language Translation (IWSLT)*.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) Demo Track*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *International Conference on Language Resources and Evaluation (LREC)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.

David Vilar. 2018. Learning hidden unit contribution for adapting neural machine translation models. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.