Specializing Multi-domain NMT via Penalizing Low Mutual Information

Jiyoung Lee†*, Hantae Kim‡, Hyunchang Cho‡, Edward Choi†, and Cheonbok Park‡

†KAIST, ‡Papago, NAVER Corp.
{jiyounglee0523, edwardchoi}@kaist.ac.kr
{hantae.kim,hyunchang.cho,cbok.park}@navercorp.com

Abstract

Multi-domain Neural Machine Translation (NMT) trains a single model with multiple domains. It is appealing because of its efficacy in handling multiple domains within one model. An ideal multi-domain NMT should learn distinctive domain characteristics simultaneously, however, grasping the domain peculiarity is a non-trivial task. In this paper, we investigate domain-specific information through the lens of mutual information (MI) and propose a new objective that penalizes low MI to become higher. Our method achieved the state-of-the-art performance among the current competitive multi-domain NMT models. Also, we empirically show our objective promotes low MI to be higher resulting in domain-specialized multi-domain NMT.

1 Introduction

Multi-domain Neural Machine Translation (NMT) (Sajjad et al., 2017; Farajian et al., 2017) has been an attractive topic due to its efficacy in handling multiple domains with a single model. Ideally, a multi-domain NMT should capture both general knowledge (e.g., sentence structure, common words) and domain-specific knowledge (e.g., domain terminology) unique in each domain. While the shared knowledge can be easily acquired via sharing parameters across domains (Kobus et al., 2017), obtaining domain specialized knowledge is a challenging task. Haddow and Koehn (2012) demonstrate that a model trained on multiple domains sometimes underperforms the one trained on a single domain. Pham et al. (2021) shows that separate domain-specific adaptation modules are not sufficient to fully-gain specialized knowledge.

In this paper, we reinterpret domain specialized knowledge from mutual information (MI) perspective and propose a method to strengthen it. Given a source sentence $X$, target sentence $Y$, and corresponding domain $D$, the MI between $D$ and the translation $Y|X$ (i.e., $MI(D; Y|X)$) measures the dependency between the domain and the translated sentence. Here, we assume that the larger $MI(D; Y|X)$, the more the translation incorporates domain knowledge. Low MI is undesirable because it indicates the model is not sufficiently utilizing domain characteristics in translation. In other words, low MI can be interpreted as a domain-specific information the model has yet to learn. For example, as shown in Fig. 1, we found that a model with low MI translates an IT term ‘computing totals’ to the vague and plain term ‘calculation’. However, once we force the model to have high MI, ‘computing totals’ is correctly retained in its translation. Thus, maximizing MI promotes multi-domain NMT to be domain-specialized.

Motivated by this idea, we introduce a new method that specializes multi-domain NMT by...
penalizing low MI. We first theoretically derive $\text{MI}(D; Y|X)$, and formulate a new objective that weights more penalty on subword-tokens with low MI. Our results show that the proposed method improves the translation quality in all domains. Also, the MI visualization ensures that our method is effective in maximizing MI. We also observed that our model performs particularly better on samples with strong domain characteristics.

The main contributions of our paper are as follows:

- We investigate MI in multi-domain NMT and present a new objective that penalizes low MI to have higher value.
- Extensive experiment results prove that our method truly yields high MI, resulting in domain-specialized model.

## 2 Related Works

### Multi-Domain Neural Machine Translation

Multi-Domain NMT focuses on developing a proper usage of domain information to improve translation. Early studies had two main approaches: injecting source domain information and adding a domain classifier. For adding source domain information, Kobus et al. (2017) inserts a source domain label as an additional tag with input or as a complementary feature. For the second approach, Britz et al. (2017) trains the sentence embedding to be domain-specific by updating using the gradient from the domain-classifier.

While previous work leverages domain information by injection or implementing an auxiliary classifier, we view domain information from MI perspective and propose a loss that promotes model to explore domain specific knowledge.

### Information-Theoretic Approaches in NMT

Mutual information in NMT is primarily used either as metrics or a loss function. For metrics, Bugliarello et al. (2020) proposes cross-mutual information (XMI) to quantify the difficulty of translating between languages. Fernandes et al. (2021) modifies XMI to measure the usage of the given context during translation. For the loss function, Xu et al. (2021) proposes bilingual mutual information (BMI) which calculates the word mapping diversity, further applied in NMT training. Zhang et al. (2022) improves the model translation by maximizing the MI between a target token and its source sentence based on its context.

Above work only considers general machine translation scenarios. Our work differs in that we integrate mutual information in multi-domain NMT to learn domain-specific information. Unlike other methods that require training of an additional model, our method can calculate MI within a single model which is more computation-efficient.

## 3 Proposed Method

In this section, we first derive MI in multi-domain NMT. Then, we introduce a new method that penalizes low MI to have high value resulting in a domain-specialized model.

### 3.1 Mutual Information in Multi-Domain NMT

Mutual Information (MI) measures a mutual dependency between two random variables. In multi-domain NMT, the MI between the domain $(D)$ and translation $(Y|X)$, expressed as $\text{MI}(D; Y|X)$, represents how much domain-specific information is contained in the translation. $\text{MI}(D; Y|X)$ can be written as follows:

$$
\text{MI}(D; Y|X) = \mathbb{E}_{D,X,Y} \left[ \log \frac{p(Y|X,D)}{p(Y|X)} \right]. \quad (1)
$$

The full derivation can be found in Appendix B. Note that the final form of $\text{MI}(D; Y|X)$ is a log quotient of the translation considering domain and translation without domain.

Since the true distributions are unknown, we approximate them with a parameterized model (Bugliarello et al., 2020; Fernandes et al., 2021), namely the cross-MI (XMI). Naturally, a generic domain-agnostic model (further referred to as \textit{general} and abbreviated as $G$) output would be the appropriate approximation of $p(Y|X)$. A domain-adapted (further shortened as DA) model output would be suitable for $p(Y|X,D)$. Hence, $\text{XMI}(D; Y|X)$ can be expressed as Eq. (2) with each model output.

$$
\text{XMI}(D; Y|X) = \mathbb{E}_{D,X,Y} \left[ \log \frac{p_{DA}(Y|X,D)}{p_{G}(Y|X)} \right] \quad (2)
$$

### 3.2 MI-based Token Weighted Loss

To calculate XMI, we need outputs from both general and domain-adapted models. Motivated by the success of adapters (Houlsby et al., 2019) in multi-domain NMT (Pham et al., 2021), we assign adapters $\phi_1, \cdots \phi_N$ for each domain ($N$ is the total number of domains). We first theoretically derive $\text{MI}(D; Y|X)$, and formulate a new objective that weights more penalty on subword-tokens with low MI. Our results show that the proposed method improves the translation quality in all domains. Also, the MI visualization ensures that our method is effective in maximizing MI. We also observed that our model performs particularly better on samples with strong domain characteristics.
number of domains) and have an extra adapter $\phi_G$ for general. We will denote the shared parameter (e.g., self-attention and feed-forward layer) as $\theta$. For a source sentence $x$ from domain $d$, $x$ passes the model twice, once through the corresponding domain adapter, $\phi_d$, and the other through the general adapter, $\phi_G$. Then, we treat the output probability from domain adapter as $p_{DA}$ and from general adapter as $p_G$. For the $i^{th}$ target token, $y_i$, we calculate XMI as in Eq. (3),

$$p(y_i | y_{<i}, x, \theta, \phi_d) - p(y_i | y_{<i}, x, \theta, \phi_G)$$ (3)

, where $y_{<i}$ is the target subword-tokens up to, but excluding $y_i$. For simplicity, we will denote Eq. (3) as XMI$(i)$. Low XMI$(i)$ means that our domain adapted model is not thoroughly utilizing domain information during translation. Therefore, we weight more on the tokens with low XMI$(i)$, resulting in minimizing Eq. (4),

$$\mathcal{L}_{MI} = \sum_{i=0}^{n_T} (1 - \text{XMI}(i)) \cdot (1 - p(y_i | y_{<i}, x, \theta, \phi_d))$$ (4)

, where $n_T$ is the number of subword-tokens in the target sentence.

The final loss of our method is in Eq. (7), where $\lambda_1$ and $\lambda_2$ are hyperparameters.

$$\mathcal{L}_{DA} = -\sum_{i=0}^{n_T} \log(p(y_i | y_{<i}, x, \theta, \phi_d))$$ (5)

$$\mathcal{L}_{G} = -\sum_{i=0}^{n_T} \log(p(y_i | y_{<i}, x, \theta, \phi_G))$$ (6)

$$\mathcal{L} = \mathcal{L}_{DA} + \lambda_1 \mathcal{L}_{G} + \lambda_2 \mathcal{L}_{MI}$$ (7)

4 Experiments

4.1 Experiment Setting

Dataset. We leverage the preprocessed dataset released by Aharoni and Goldberg (2020) consisting of five domains (IT, Koran, Law, Medical, Subtitles) available in OPUS (Tiedemann, 2012; Aulamo and Tiedemann, 2019). More details on the dataset and preprocessing are described in Appendix A.

Baseline. We compare our method with the following baseline models: (1) Mixed trains a model on all domains with uniform distribution, (2) Domain-Tag (Kobus et al., 2017) inserts domain information as an additional token in the input, (3) Multitask Learning (MTL) (Britz et al., 2017) trains a domain classifier simultaneously and encourage the sentence embedding to encompass its domain characteristics, (4) Adversarial Learning (AdvL) (Britz et al., 2017) makes the sentence embedding to be domain-agnostic by flipping the gradient from the domain classifier before the back-propagation, (5) Word-Level Domain Context Discrimination (WDC) (Zeng et al., 2018) integrates two sentence embedding which are trained by MTL and AdvL respectively, (6) Word-Adaptive Domain Mixing (Jiang et al., 2020), has domain-specific attention heads and the final representation is the combination of each head output based on the predicted domain proportion, and (7) Domain-Adapter (Pham et al., 2021) has separate domain adapters (Houlsby et al., 2019) and a source sentence passes through its domain adapters. This can be regarded as our model without general adapter and trained with $\mathcal{L}_{DA}$.

4.2 Main Results

Table 1 presents sacreBLUE (Post, 2018) and chrF (Popović, 2015) score from each model in all domains. For a fair comparison, we matched the number of parameters for all models. Baseline results following its original implementation with different parameter size are provided in Appendix C. Interestingly, Mixed performs on par with Domain-Tag and outperforms Word-Adaptive Domain Mixing, suggesting that not all multi-domain NMT methods are effective. Although adapter-based models (i.e., Ours (w/o $\mathcal{L}_{MI}$) and Domain-Adapter) outperform Mixed, the performance increase is still marginal. Our model has gained 1.15 BLEU improvement over Mixed. It also outperforms all baselines with statistically significant difference.

As an ablation study of our MI objective, we conduct experiments without $\mathcal{L}_{MI}$ to prove its effectiveness. The result confirms that $\mathcal{L}_{MI}$ encouraged the model to learn domain specific knowledge leading to refined translation.

4.3 Mutual Information Distribution

We visualize XMI$(i)$ in Eq. (3) to verify that our proposed loss penalizes low XMI. Figure 2 is the histogram of XMI$(i)$ from the test samples in Law. Other domain distributions are in Appendix D. We use Domain-Adapter for comparison since it performs the best among the baselines. For $p_G$, we use the output probability of Mixed for both cases. From the distributions, our method indeed penal-
43.87 ± 0.505 20.31 ± 0.371 58.33 ± 0.474 55.19 ± 0.737 30.36 ± 0.424 41.61 43.86 ± 0.167 20.33 ± 0.275 58.40 ± 0.195 55.36 ± 0.243 30.43 ± 0.367 41.71 44.44 ± 0.193 20.75 ± 0.212 58.49 ± 0.193 55.43 ± 0.308 30.52 ± 0.242 41.93 41.88 ± 0.240 19.84 ± 0.297 55.82 ± 0.594 52.88 ± 0.783 30.39 ± 0.141 40.16 60.37 ± 0.113 41.02 ± 0.212 71.79 ± 0.290 67.62 ± 0.306 45.63 ± 0.113 57.29 44.50 ± 0.342 20.37 ± 0.193 58.22 ± 0.169 56.00 ± 0.243 31.02 ± 0.334 42.02 62.30 ± 0.248 41.65 ± 0.160 73.40 ± 0.066 69.54 ± 0.149 46.30 ± 0.306 58.64

Table 2 reports BLEU score improvement compared to Domain-Adapter in each quartile along with averaged performance increases. In Law, Medical and Subtitles, BLEU score improvement increases as quartile gets higher. Furthermore, the improvements in Q2-Q3 and Q3-Q4 are larger than the averaged improvement score (i.e., Average column). However, in IT and Koran, -Q1 has the largest performance increases. We conjecture the reason is that in both domains, the number of

Table 1: Average and standard deviation of BLEU (upper line) and chrF (bottom line) from five random seed sentences in each domain (examples are in Appendix E) and consider them as domain-specific keywords. We assume that the more these keywords are included in the source sentence, the more domain specialized the sample is. We divide the test set into quartiles based on the number of the keywords the source sentence contains.

|                           | IT              | Koran           | Law             | Medical         | Subtitles        | Average   |
|---------------------------|-----------------|-----------------|-----------------|-----------------|------------------|----------|
| Mixed                     | 43.87 ± 0.505   | 20.31 ± 0.371   | 58.33 ± 0.474   | 55.19 ± 0.737   | 30.36 ± 0.424    | 41.61    |
| Domain-Tag                | 44.29 ± 0.142   | 20.44 ± 0.236   | 58.47 ± 0.275   | 55.39 ± 0.288   | 30.61 ± 0.220    | 41.84    |
| MTL                       | 44.00 ± 0.208   | 20.40 ± 0.198   | 58.27 ± 0.327   | 55.24 ± 0.564   | 30.52 ± 0.478    | 41.69    |
| AdvL                      | 43.86 ± 0.167   | 20.33 ± 0.275   | 58.40 ± 0.195   | 55.36 ± 0.243   | 30.43 ± 0.367    | 41.71    |
| WDC                       | 44.44 ± 0.193   | 20.75 ± 0.212   | 58.49 ± 0.193   | 55.43 ± 0.308   | 30.52 ± 0.242    | 41.93    |
| Word-Adaptive Domain Mixing | 41.88 ± 0.240   | 19.84 ± 0.297   | 55.82 ± 0.594   | 52.88 ± 0.783   | 30.39 ± 0.141    | 40.16    |
| Domain-Adapter            | 44.50 ± 0.342   | 20.37 ± 0.193   | 58.22 ± 0.169   | 56.00 ± 0.243   | 31.02 ± 0.334    | 42.02    |
| Ours (w/o ΔMI)            | 44.65 ± 0.318   | 20.43 ± 0.286   | 58.21 ± 0.692   | 55.38 ± 0.664   | 30.82 ± 0.498    | 41.90    |
| Ours                      | 45.89 ± 0.215   | 20.80 ± 0.298   | 59.22 ± 0.306   | 56.34 ± 0.238   | 31.56 ± 0.218    | 42.76    |
|                           | (+1.39) (+0.43) | (+1.00) (+0.34) | (+0.54) (+0.74) | (+0.89) (+0.39) | (+0.62) (+0.16) | (+0.49) |

Figure 2: XMI distribution in Law. X-axis is XMI and Y-axis is the density. Green is Domain-Adapter and blue is our model. Our model has more high XMI values. 

4.4 Translation Performance for Domain Specialized Sentences

Since the ultimate goal is to specialize multi-domain NMT, we calculate BLEU score improvement according to the domain specificity. We extract top 1% TF-IDF words in train source sentences in each domain (examples are in Appendix E) and consider them as domain-specific keywords. We assume that the more these keywords are included in the source sentence, the more domain specialized the sample is. We divide the test set into quartiles based on the number of the keywords the source sentence contains.

Table 2: BLEU improvements in each quartiles. Q1, Q2, Q3 and Q4 represents 25%, 50%, 75% 100% respectively. The higher the quartile, the more domain specific the samples.
Table 3: Average and standard deviation of BLEU (upper line) and chrF (bottom line) from five random seeds on Ko-En dataset. Bold indicates the best performance within a domain.

| Domain-Adapter | Finance | Ordinance | Tech | Average |
|----------------|---------|-----------|------|---------|
| Mixed          | 72.64±0.105 | 75.38±0.091 | 81.64±0.105 | 76.60±0.105 |
| Domain-Tag     | 52.71±0.221 | 56.60±0.115 | 66.03±0.190 | 58.45±0.450 |
| WDC            | 72.78±0.128 | 75.34±0.105 | 81.53±0.105 | 76.55±0.185 |
| Domain-Adapter | 53.13±0.166 | 56.97±0.115 | 66.25±0.115 | 58.78±0.191 |
| Ours           | 53.87±0.166 | 57.47±0.115 | 66.66±0.115 | 59.33±0.233 |

Figure 3: Example visualizations with MI values from IT and Medical. The more intense the red, the more higher MI value.

Table 4: Comparison of training computation cost between Domain-Adapter and Ours. The values are averaged across five seed experiments.

| Domain-Adapter | Ours |
|----------------|------|
| Number of Iterations (↑) | 51.7K |
| Peak Memory (GB) (↓) | 27.09 |
| Words per Second (↑) | 22.9K |
| Updates per Second (↑) | 0.4 |

4.5 Experiment Results on Korean-English

To verify the effectiveness of our proposed method in different language, we additionally conducted experiment on Korean-English dataset which has approximately 1M samples with three domains: Finance, Ordinance and Tech. The dataset is obtained from Alhub\(^2\) which is publicly available. Model configuration is identical with the main experiment on OPUS. More experimental details are in Appendix A.

Table 3 demonstrates the results from the major baselines and our model, where we select top-4 baselines in the experiment on OPUS (Table 1). Our model achieves the best performance in all three domains, outperforming Domain-Adapter by 0.55 BLEU score on average. This result confirms that our proposed method can be further extended to other languages.

4.6 Samples with MI Visualization

Figure 3 demonstrates test set outputs with MI values generated by our model. Color intensity is correlated with MI value; the more intense the red, the more higher MI value. Note that the model has high MI especially when generate domain-specific words (e.g., ‘password’ in IT and ‘omalizumab’ in Medical). This result is analogous to our motivation in that high MI value encourages model to translate domain peculiar terms. More samples are provided in Appendix F.

4.7 Computation Cost for Training

We compare computation cost between Domain-Adapter and our proposed model. Domain-Adapter was chosen because it shares the same model architecture with ours. Table 4 provides four computation cost during training: Number of Iterations until converge, Peak Memory, Words per Second, and Updates per Second. Our model requires fewer number of iterations needed to be trained (3.2K difference on average across five seeds). Our model has slightly higher peak memory (0.3GB, 0.94%) than Domain-Adapter, however, we believe this is acceptable when considering the performance improvement. Furthermore, there was not much difference in words per second and updates per second during training.

5 Conclusion

We build a specialized multi-domain NMT by adding MI-based loss. We reinterpret domain-specific knowledge from MI perspective and promote a model to explore domain knowledge by penalizing low MI. Our results prove that the proposed method is effective in increasing overall MI.
Limitations

Although many previous multi-domain NMT studies regard the source of the given sentence as its domain, equating domain and corpora is a naive approach and can partially represent the data. (Aharoni and Goldberg, 2020) For instance, some sentences may incorporate multiple domain characteristics or can be better translated under different domain other than its source domain. (Currey et al., 2020; Pham et al., 2021) This problem is not limited to our work but is applicable to other previous multi-domain NMT studies. Establishing a more proper definition of domain is a future work and a critical challenge in multi-domain NMT.

Acknowledgement

The authors would like to thank all members in Papago Team, NAVER Corporation for their valuable comments. Also, we sincerely thank Jin-Hwa Kim at NAVER AI Lab for the insightful feedback. This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)), and National Research Foundation of Korea (NRF) grant (NRF-2020H1D3A2A03100945) funded by the Korea government (MSIT).

References

Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pages 7747–7763.

Mikko Aulamo and Jörg Tiedemann. 2019. The OPUS resource repository: An open package for creating parallel corpora and machine translation services. In Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa), pages 389–394.

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. arXiv preprint arXiv:1909.08478.

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In Proceedings of the Second Conference on Machine Translation (WMT), pages 118–126.

Emanuele Bugliarello, Sabrina J. Mielke, Antonios Anastasopoulos, Ryan Cotterell, and Naoaki Okazaki. 2020. It’s easier to translate out of English than into it: Measuring neural translation difficulty by cross-mutual information. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pages 1640–1649.

Anna Currey, Prashant Mathur, and Georgiana Dinu. 2020. Distilling multiple domains for neural machine translation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4500–4511.

M Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In Proceedings of the Second Conference on Machine Translation (WMT), pages 127–137.

Patrick Fernandes, Kayo Yin, Graham Neubig, and André F. T. Martins. 2021. Measuring and increasing context usage in context-aware machine translation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL), pages 6467–6478.

Barry Haddow and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on smt systems. In Proceedings of the Seventh Workshop on Statistical Machine Translation (SMT), pages 422–432.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In International Conference on Machine Learning (ICML), pages 2790–2799.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2020. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pages 1823–1834.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In Proceedings of the International Conference Recent Advances in Natural Language Processing, (RANLP), pages 372–378.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion
Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations) (NAACL), pages 48–53.

MinhQuang Pham, Josep Maria Crego, and François Yvon. 2021. Revisiting multi-domain machine translation. Transactions of the Association for Computational Linguistics (TACL), pages 17–35.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In Proceedings of the Tenth Workshop on Statistical Machine Translation (SMT), pages 392–395.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In Proceedings of the Third Conference on Machine Translation: Research Papers (WMT), pages 186–191.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel. 2017. Neural machine translation training in a multi-domain scenario. arXiv preprint arXiv:1708.08712.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), pages 1715–1725.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC), pages 2214–2218.

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 (NIPS).

Yangyifan Xu, Yijin Liu, Fandong Meng, Jiajun Zhang, Jinan Xu, and Jie Zhou. 2021. Bilingual mutual information based adaptive training for neural machine translation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL), pages 511–516.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 447–457.

Songming Zhang, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, Jian Liu, and Jie Zhou. 2022. Conditional bilingual mutual information based adaptive
A Experiment Settings

A.1 Dataset
For preprocessing, we conducted tokenization and normalize-punctuation by using Moses (Koehn et al., 2007) pipeline. We additionally eliminated samples with (i) sequences shorter than one subword-token, (ii) sequences longer than 250 subword-tokens, (iii) severe length imbalance between the language pair (top, bottom 5% for each domain) for both De-En and Ko-En. Table 5 and 6 show the final number of samples.

| Domain   | De-En Train | Dev | Test |
|----------|-------------|-----|------|
| IT       | 211,374     | 1,888 | 2,000 |
| Koran    | 16,952      | 1,872 | 2,000 |
| Law      | 434,555     | 1,861 | 2,000 |
| Medical  | 233,167     | 1,873 | 2,000 |
| Subtitles| 470,611     | 1,899 | 2,000 |

Table 5: Number of samples in De-En

| Domain   | Ko-En Train | Dev | Test |
|----------|-------------|-----|------|
| Finance  | 156,569     | 9,510 | 5,000 |
| Ordinance| 79,802      | 9,335 | 5,000 |
| Tech     | 711,885     | 9,251 | 5,000 |

Table 6: Number of samples in Ko-En

A.2 Experiment Details
For De-En, we use a joint BPE vocabulary (Senrich et al., 2016) learned with 32k merge operations. For Ko-En, we train BPE vocabulary with 32k size separately for each language since Korean and English do not share characters. Remaining experiment settings are identical for both language pairs. Our experiments are conducted under open-source fairseq\(^3\) (Ott et al., 2019) framework. We built upon Transformer model (Vaswani et al., 2017) which has 6 encoder and decoder layers with embedding dimension of 512, feed-forward dimension of 2048, and attention heads of 8. Parameters of encoder embedding, decoder embedding and decoder last layer are shared. We also utilize the same sinusoidal positional embedding following the original work. We fix dropout to 0.1 and used ReLU (Agarap, 2018) as an activation function. Following Bapna et al. (2019), the domain and general adapters \((\phi_1, \cdots, \phi_N, \phi_G)\) are inserted after feed-forward layer following the multi-head attention. Note that our training differs in that we jointly train all parameters from scratch including adapters on all domains.

The bottleneck size of the adapters is 256. Adapters are initialized from zero-mean Gaussian with standard deviation \(10^{-2}\) which proven to be most effective in the proposed work. We searched the best combination of \(\lambda_1\) and \(\lambda_2\) by grid search ranging from 0.5 to 1.0. Then, we set \(\lambda_1\) and \(\lambda_2\) both to 1. All experiments are trained with label-smoothing cross-entropy loss with smoothing parameter of 0.1. All experiments are conducted using 8 NVIDIA V100 GPU.

In all experiments, a model is trained until early stopping with patience of 10 based on BLEU. We use Adam (Kingma and Ba, 2014) optimizer with an initial learning rate of \(5 \cdot e^{-4}\), where the learning rate is searched within the range of 0.001 to 0.0001. The tokens per batch is 8192 in all experiments. We compute sacreBLEU score\(^4\) from outputs using beam search with a beam size of 5.

A.3 Baselines
In this section, we provide a detailed explanation on each baseline. Mixed and Domain-Tag employ identical model architecture with our model excluding adapters. Mixed does not utilize domain information and treat all samples are from identical distribution. On the other hand, Domain-Tag distinguishes domain by adding domain tag in front of the input sentence. We enlarged Mixed and Domain-Tag to Mixed-Big and Domain-Tag-big by increasing encoder and decoder embedding dimension to 608.

For Word-Adaptive Domain Mixing, we borrowed publicly available code and applied on our dataset. We applied word-adaptive training in both encoder and decoder because it had the best performance in the original paper. Other configurations are the same with ours.

Domain-Adapter has the domain-specific adapters \(\phi_1, \cdots, \phi_N\), and the input sentence passes through only its domain adapter. Note that there are two major differences between Domain-Adapter and ours. First, Domain-Adapter does not need general adapter \(\phi_G\) since it does not calculate \(p_G(Y|X)\). Second, a source sentence

\(^3\)https://github.com/facebookresearch/fairseq

\(^4\)sacreBLEU signature: BLEU+c.mixed+l.de-en+#.1+s.exp+tok.13a+v.2.0.0
only passes through the model once only through its domain adapter $\phi_d$. Similar to our method, all the parameters including adapters are jointly trained from scratch.

**B Full Derivation of Domain-Aware Mutual Information**

Below is the full derivation of Domain-Aware Mutual Information.

\[
MI(D; Y|X) = \mathbb{E}_{D,X,Y} \left[ \log \frac{p(D,Y|X)}{p(D|X) \cdot p(Y|X)} \right] \\
= \mathbb{E}_{D,X,Y} \left[ \log \frac{p(D|Y,X) \cdot p(Y|X)}{p(D|X) \cdot p(Y|X)} \right] \\
= \mathbb{E}_{D,X,Y} \left[ \log \frac{p(X,Y,D)}{p(X,Y) \cdot p(X,D)} \right] \\
= \mathbb{E}_{D,X,Y} \left[ \log \frac{p(Y|X,D)}{p(Y|X)} \right]
\]

The proof from the first to the second line is provided below.

\[
P(X, Y, D) = P(D|Y, X) \cdot P(Y|X) \cdot P(X) \\
\Rightarrow P(D, Y|X) = P(D|Y, X) \cdot P(Y|X)
\]

**C Additional Baseline Results**

Table 7 shows sacreBLEU (Post, 2018) and chrF (Popović, 2015) score from baseline models (Mixed-Small, Domain-Tag-Small, Word-Adaptive Domain Mixing Big) following its original implementation with different number of parameters. Note that Mixed and Domain-Tag underperform the models in the main experiment, and Word-Adaptive Domain Mixing becomes effective when enlarge model size.

**D MI Histogram**

XMI histograms from all doamins are in Fig. 4. Adpater (baseline) is colored in green and our model is in blue. From the plot, we can verify that in all domains, our model outputs higher XMI compared to the baseline.

**E Details on Translation Performance for Domain Specialized Experiment**

Examples of extracted TF-IDF keywords are in Table 8. We removed stop words and conducted lemmatization before extracting keywords. As expected, chosen words are correlated to its domain are chosen.
| # of Parameters | IT  | Koran | Law  | Medical | Subtitles | Average |
|-----------------|-----|-------|------|---------|-----------|---------|
| Mixed-Small     | 60M | 43.64±0.253 | 20.74±0.155 | 57.47±0.376 | 54.88±0.353 | 30.58±0.273 | 41.46 |
|                 |     | 61.96±0.204 | 42.01±0.159 | 72.95±0.275 | 68.98±0.362 | 46.06±0.293 | 58.39 |
| Mixed           | 76M | 43.87±0.505 | 20.31±0.371 | 58.33±0.474 | 55.19±0.737 | 30.36±0.424 | 41.61 |
|                 |     | 62.00±0.403 | 41.75±0.343 | 73.41±0.303 | 69.14±0.346 | 45.73±0.424 | 58.40 |
| Domain-Tag-Small| 60M | 44.00±0.409 | 20.39±0.251 | 57.80±0.206 | 54.91±0.197 | 31.10±0.136 | 41.64 |
|                 |     | 62.09±0.242 | 41.71±0.141 | 73.14±0.183 | 68.97±0.056 | 46.34±0.209 | 58.45 |
| Domain-Tag      | 76M | 44.29±0.142 | 20.44±0.236 | 58.47±0.273 | 55.39±0.288 | 30.61±0.220 | 41.84 |
|                 |     | 62.30±0.111 | 41.75±0.203 | 73.56±0.190 | 69.28±0.160 | 45.99±0.268 | 58.58 |
| MTL             | 76M | 44.00±0.298 | 20.40±0.198 | 58.27±0.327 | 55.24±0.364 | 30.52±0.478 | 41.69 |
|                 |     | 62.11±0.169 | 41.78±0.174 | 73.42±0.197 | 69.16±0.235 | 45.87±0.116 | 58.47 |
| AdvL            | 76M | 43.86±0.167 | 20.33±0.275 | 58.40±0.193 | 55.56±0.245 | 30.43±0.367 | 41.71 |
|                 |     | 61.91±0.099 | 41.79±0.206 | 73.42±0.193 | 69.30±0.184 | 45.80±0.208 | 58.44 |
| WDC             | 76M | 44.44±0.193 | 20.75±0.212 | 58.49±0.193 | 55.43±0.308 | 30.52±0.242 | 41.93 |
|                 |     | 62.27±0.175 | 42.05±0.198 | 73.58±0.182 | 69.20±0.203 | 45.87±0.125 | 58.59 |
| Word-Adaptive   | 218M| 44.08±0.561 | 20.34±0.257 | 59.63±0.308 | 56.81±0.386 | 29.34±0.488 | 42.04 |
| Domain Mixing   |     | 61.93±0.273 | 41.23±0.334 | 74.26±0.196 | 69.96±0.245 | 44.62±0.400 | 58.40 |
| Word-Adaptive   | 76M | 41.88±0.240 | 19.84±0.297 | 55.82±0.594 | 52.88±0.785 | 30.39±0.141 | 40.16 |
| Domain Mixing   |     | 60.37±0.113 | 41.02±0.212 | 71.79±0.290 | 67.62±0.396 | 45.63±0.113 | 57.29 |
| Domain-Adapter  | 76M | 44.50±0.342 | 20.37±0.193 | 58.22±0.169 | 56.00±0.243 | 31.02±0.334 | 42.02 |
|                 |     | 62.50±0.238 | 41.65±0.160 | 73.40±0.066 | 69.54±0.149 | 46.30±0.306 | 58.64 |
| Ours (w/o $\mathcal{L}_{MI}$) | 76M | 44.65±0.318 | 20.43±0.286 | 58.21±0.692 | 55.38±0.684 | 30.82±0.498 | 41.90 |
|                 |     | 62.49±0.221 | 41.77±0.262 | 73.40±0.418 | 69.28±0.377 | 46.16±0.414 | 58.62 |
| Ours            | 76M | 45.89±0.218 | 20.80±0.298 | 59.22±0.306 | 56.34±0.238 | 31.56±0.218 | 42.76 |
|                 |     | (+1.39) | (+0.43) | (+1.00) | (+0.34) | (+0.54) | (+0.74) |
|                 |     | 63.19±0.204 | 42.05±0.274 | 74.02±0.219 | 69.94±0.238 | 46.46±0.261 | 59.13 |
|                 |     | (+0.89) | (+0.39) | (+0.62) | (+0.40) | (+0.16) | (+0.49) |

Table 7: Average and standard deviation of BLEU (upper line) and chrF (bottom line) from baselines with different parameter size and our model.

Figure 4: XMI distributions from all domains. X-axis is XMI and Y-axis is the density. Adapter is colored in green and our model is in blue. Our XMI distribution has more higher values than baseline.
### Examples of TF-IDF Keywords

| Domain       | Examples                                                                 |
|--------------|---------------------------------------------------------------------------|
| IT           | übertragungsrate (transfer rate), zwischensumme (temporally save), YouTube, übertragungsfortschritt (transfer progress), speichergröße (memory size), yahoomail, zusammensetzungswerkzeug (composition tool), übersichtsmodus (overview mode), webserver, zwischengespeichert (cached), übersetzungsprogramm (translation game), zwischenablagename (clipboard name) |
| Koran        | übertreter (transgressor), zwingherr (tyrant), widmest (dedicate), unterwürfig (submissive), unterworfen (subjected), städte (cities), sterben (die), schutzherr (patron), religion, muslime (muslims) |
| Law          | überwachungszollstelle (supervising customs office), änderungsverfahren (change procedure), zustellungsmängel (delivery defects), wirtschaftsjahre (fiscal years), überstunden (overtime), zuschusssatz (subsidy rate), widerklänge (echoes), übernahmeprotokoll (takeover protocol), zulassungsvoraussetzungen (admission requirements), verwaltungskommission (administrative commission), tarife (rates) |
| Medical      | überlebenswahrsccheinlichkeit (probability of survival), verletzung (injury), tagesgesamtdosis (total daily dose), überempfindlichkeitsreaktionen (hypersensitivity reactions), zäpfchen (suppository), tremor, zytotoxisch (cytotoxic), urin (urine), Schatzungen (estimates), wirkstoffmatrix (active ingredient matrix), vorsichtsmaßnahmen (precautions) |
| Subtitles    | übungen (exercises), wähle (choose), übersehen (overlook), öffentlichkeitsarbeit (public relation), äußern (to express), ärger (trouble), ältester (oldest), zähflüssig (viscous), zwischenmahlezen (snacks), wäsche (laundry), werbepause (commercial break) |

Table 8: Examples of TF-IDF extracted words of the source language (i.e., German). We randomly sampled 11 words from each domain among the top 1% keywords. We also write its meaning in English words in the bracket.

| Domain | IT | Koran | Law | Medical | Subtitles |
|--------|----|-------|-----|---------|-----------|
| -Q1    | 0  | 0     | 1.91| 0.96    | 1.70      |
| Q1-Q2  | 1  | 1     | 5.07| 4.05    | 3.57      |
| Q2-Q3  | 2  | 2     | 8.36| 6.83    | 5.42      |
| Q3-Q4  | 4.10| 3.42  | 13.21| 11.50   | 8.11      |

Table 9: Averaged number of captured top TF-IDF keywords in each quartile.
MI Value | -1.0 | -0.71 | -0.43 | -0.14 | 0.14 | 0.43 | 0.71 | 1.0

**Domain: IT**
- **Source**: Möchten Sie diesen Zugang wirklich löschen?
- **Reference**: Do you really want to delete this account?
- **Hypothesis**: Do you really want to delete this account?
- **Source**: Datei, die eine Liste zu druckender Schriftarten enthält
- **Reference**: File containing list of fonts to print
- **Hypothesis**: File containing a list of fonts to print
- **Source**: Geben Sie IP @@ Adresse und Port des Servers ein:
- **Reference**: Enter server IP address and port:
- **Hypothesis**: Enter IP address and port of the server:

**Domain: Law**
- **Source**: Der Wortlaut der Schreiben ist dieser Verordnung beigefügt.
- **Reference**: The text of the letters is annexed to this Regulation.
- **Hypothesis**: The text of the letters is annexed to this Regulation.
- **Source**: Inkrafttreten, Änderung und Kündigung des Abkommens
- **Reference**: Entry into force, amendments to and termination of the Agreement
- **Hypothesis**: Entry into force, amendment and termination of the Agreement
- **Source**: Im Namen der Gemeinschaft Für die Republik Bulgarien
- **Reference**: On behalf of the Community For the Republic of Bulgaria
- **Hypothesis**: On behalf of the Community For the Republic of Bulgaria

**Domain: Medical**
- **Source**: Stabilität der rekonstituierten Suspension im Infusionsbeutel:
- **Reference**: Stability of the reconstituted suspension in the infusion bag:
- **Hypothesis**: Stability of the reconstituted suspension in the infusion bag:
- **Source**: Jede Durchstechflasche enthält 10 mg Basiliximab *.
- **Reference**: Each vial contains 10 mg basiliximab *.
- **Hypothesis**: Each vial contains 10 mg of basiliximab *.
- **Source**: Bei älteren Patienten wurden keine pharmakokinetischen Studien durchgeführt.
- **Reference**: Pharmacokinetic studies have not been performed in the elderly.
- **Hypothesis**: No pharmacokinetic studies have been performed in the elderly.

Figure 5: Visualization of examples with MI values from IT, Law, and Medical. Color intensity is correlated with MI value; the more intense the red, the more higher MI value.