Abstract

Neural machine translation (NMT) models do not work well in domains different from the training data. The standard approach to this problem is to build a small parallel data in the target domain and perform domain adaptation from a source domain where massive parallel data is available. However, domain adaptation between distant domains (e.g., subtitles and research papers) does not perform effectively because of mismatches in vocabulary; it will encounter many domain-specific unknown words (e.g., ‘angstrom’) and words whose meanings shift across domains (e.g., ‘conductor’). In this study, aiming to solve these vocabulary mismatches in distant domain adaptation, we propose vocabulary adaptation, a simple method for effective fine-tuning that adapts embedding layers in a given pre-trained NMT model to the target domain. Prior to fine-tuning, our method replaces word embeddings in embedding layers of the NMT model, by projecting general word embeddings induced from monolingual data in the target domain onto the source-domain embedding space. Experimental results on distant domain adaptation for English-to-Japanese translation and German-to-English translation indicate that our vocabulary adaptation improves the performance of fine-tuning by 3.6 BLEU points.

1 Introduction

Although neural machine translation (NMT) has achieved the state-of-the-art translation performance, the performance of NMT models remarkably drops in domains different from the training data (Koehn and Knowles, 2017). Since massive parallel data is available in a few domains, domain adaptation between distant domains is often required to employ NMT in practical applications.

Assuming a small target-domain and massive source-domain parallel data, researchers have developed two generic approaches to supervised domain adaptation for NMT: fine-tuning (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016; Chu et al., 2017; Thompson et al., 2018; Khayrallah et al., 2018; Bapna and Firat, 2019) and multi-domain learning (Kobus et al., 2017; Britz et al., 2017; Wang et al., 2017) (§ 2). Fine-tuning adjusts parameters of a model pre-trained from the source-domain parallel data using a small target-domain parallel data, while multi-domain learning induces a model from scratch using both source- and target-domain parallel data. However, when two domains differ substantially, vocabulary mismatches cause serious problems to perform domain adaptation effectively; namely, a model can handle neither domain-specific words that are not covered in the small target-domain parallel data (unknown word problem) nor words that have different meanings across domains (semantic shift).

To resolve these vocabulary-mismatch problems in distant domain adaptation, we propose vocabulary adaptation (Figure 1), a method of directly adapting the vocabulary (and the embedding layers) of a pre-trained NMT model to the target domain, to perform effective fine-tuning (§ 3).
key idea behind our approach is to regard source and target domains as independent languages/tasks, and utilize methods for inducing cross-lingual/task word embeddings to bridge word embeddings in the source and target domains. Given an NMT model pre-trained in the source domain, we first induce wide-coverage target-domain word embeddings from target-domain monolingual data. We then fit the obtained target-domain word embeddings to the embedding space of the pre-trained NMT model, by inducing a cross-domain projection from the target-domain embedding space to the source-domain embedding space. To perform this cross-domain embedding projection, we explore two methods: orthogonal linear transformation (Xing et al., 2015) and locally linear mapping (LLM) (Sakuma and Yoshinaga, 2019).

We evaluated fine-tuning with our vocabulary adaptation for two domain pairs: 1) from JESC (Pryzant et al., 2018) to ASPEC (Nakazawa et al., 2016) on English to Japanese translation, and 2) from IT domain to Medical domain of the dataset constructed by (Koehn and Knowles, 2017) on English to German translation (hereafter, En→Ja and En→De, respectively) (§ 4). Experimental results demonstrated that our domain adaptation method improved BLEU scores on distant domain adaptation (En→Ja) by 3.60 points (21.70 to 25.30) compared to traditional fine-tuning (Luong and Manning, 2015) and shows further improvements by 3.92 points (25.15 to 29.07) in terms of BLEU score (Papineni et al., 2002) when combining with back-translation (Sennrich et al., 2016a).

The contribution of this paper is as follows:

- We established an effective distant domain adaptation for NMT by adapting vocabulary (and their embeddings) of a pre-trained NMT model to the target domain prior to fine-tuning.
- We confirmed the limitation of linear transformation for cross-domain embedding projection due to differences in the embedding topologies across domains.

We will release all the codes to promote the reproducibility of our results.

2 Related Work

There are two common approaches to domain adaptation for NMT: fine-tuning and multi-domain learning. In what follows, we first review these domain adaptation approaches, and then introduce related work on solving unknown word problems in NMT. **Multi-domain learning** induces an NMT model from parallel data in both source and target domains (Kobus et al., 2017; Wang et al., 2017; Britz et al., 2017). Since this approach requires the models to handle both the source- and target-domain data, the training cost becomes problematic when we perform adaptation to many target domains.

**Fine-tuning** (alternatively referred to as continued learning) is a lightweight domain adaptation method. Given an NMT model pre-trained with massive parallel data in the source domain (pre-training), it retrains this pre-trained model with a small parallel data in the target domain (Luong and Manning, 2015; Chu et al., 2017; Thompson et al., 2018; Bapna and Firat, 2019; Gu et al., 2019).

Recent studies focus on the model’s architecture (which parameters to update), the training objective, or the strategy in training. Meanwhile, few attempts have been made to resolve the vocabulary mismatch problem between distant domains.

**Unsupervised domain adaptation** exploits target-domain monolingual data to learn a target-domain language model that promotes to generate natural sentences in the target domain! (Gülçehre et al., 2015; Domhan and Hieber, 2017). Data augmentation using back-translation (Sennrich et al., 2016a) is another approach to exploit the target-domain monolingual data. Although these approaches can partly address the problem of semantic shift, they assume that the vocabulary of models trained from source-domain parallel data covers the vocabulary of the target domain.

**Handling unknown words** has been extensively studied in NMT, since the vocabulary size of an NMT model is limited due to practical requirements (e.g., GPU memory). Early approaches exploit a bilingual dictionary to replace unknown words in the decoder outputs (Jean et al., 2015; Luong et al., 2015). The current standard approach to the unknown word problem is to use token units shorter than words such as characters (Ling et al., 2015; Luong and Manning, 2016) and subwords (Sennrich et al., 2016b; Kudo, 2018) to handle rare words as a sequence of known characters (subwords). However, it is possible that the semantic shift also occurs in characters or subwords, which hinders domain adaptation.

In this study, we aim to provide pre-trained NMT models with a functionality that directly handles
both target-domain-specific unknown words and semantic shift across domains, by exploiting cross-domain embeddings learned from target-domain monolingual data.

3 Vocabulary Adaptation for Distant Domain Adaptation in NMT

As we have discussed so far (§ 2), vocabulary mismatches between the source and target domains are the major challenge in distant domain adaptation for NMT. This section proposes a method of directly resolving this problem by exploiting target-domain monolingual data in the fine-tuning-based domain adaptation. Although our method is applicable to any NMT model with embedding layers, we assume here standard encoder-decoder models (Bahdanau et al., 2015; Vaswani et al., 2017) for clarity.

In fine-tuning, given an NMT model pre-trained in the source domain, we continue to train the model’s parameters using a small parallel data in the target domain. Since the vocabulary in the pre-trained model is constructed in the source domain, it does not cover domain-specific words in the target domain. Even for words shared across both domains, their embeddings in the pre-trained NMT model represent the meanings in the source domain which can be different from ones in the target domain (e.g., ‘conductor’ in subtitle domain vs. in the scientific domain; ‘conductor’ can refer to a coil instead of musical conductor).

To mitigate this problem, we propose to replace the encoder’s embeddings, the decoder’s embeddings, and the output layer of the pre-trained model with word embeddings induced from target-domain monolingual data. However, as in transplanting organs from a donor to a recipient, this causes rejection; the embedding space of the pre-trained model is irrelevant to the space of the target-domain word embeddings. We therefore project the target domain word embeddings onto the semantic space of the pre-trained model in order to make the target-domain word embeddings compatible with the pre-trained model (Figure 1). This approach is inspired by cross-lingual word embeddings that bridge word embeddings across languages/tasks.

The overview of our proposed method is summarized as follows.

Step 0 (Pre-training) We assume an NMT model pre-trained from massive source-domain data. Note that this model is intended to be adapted to various target domains. The vocabulary is not tailored to a specific target domain.

Step 1 (Inducing target-domain embeddings) We induce word embeddings from monolingual data in the target domain for both languages. Although we can use any method for induction, we adopt Continuous Bag-of-Words (CBOW) (Mikolov et al., 2013) here since CBOW is effective for initializing embeddings in NMT (Neishi et al., 2017).

Step 2 (Embedding projection) We project the target-domain embeddings of the source and target languages into the embedding spaces of the pre-trained encoder and decoder, respectively, to obtain cross-domain embeddings.

Step 3 (Fine-tuning) We replace the vocabularies and the embedding layers by the cross-domain embeddings, and apply fine-tuning using the target-domain parallel data.

To acquire cross-domain embedding projection, we regard two domains as languages and tasks and explore two methods for inducing cross-lingual (Xing et al., 2015) and cross-task word embeddings (Sakuma and Yoshinaga, 2019). In what follows, we explain each method in detail.

3.1 Vocabulary Adaptation by Linear Transformation

The first method exploits an orthogonal linear transformation (Xing et al., 2015) to obtain cross-lingual word embeddings. We use words shared across two domains to prepare a bilingual dictionary, and then induce an orthogonal linear transformation from the embeddings of the target domain to the embeddings of the source domain. The obtained linear transformation is used to map the entire embeddings of the target domains to the embeddings space of the source domain.

3.2 Vocabulary Adaptation by Locally Linear Mapping

Due to the difference in the domains and tasks (CBOW and NMT) to induce the embeddings, the linear transformation is likely to fail. Thus, we employ a recent method for cross-task embedding projection called locally linear mapping (LLM) (Sakuma and Yoshinaga, 2019). The LLM learns a projection that preserves the local topology
(positional relationships) of the original embeddings after mapping while disregarding the global topology. This property of LLM is suited to our situation because the local topology is expected to be the same across the semantic spaces of the two domains, while globally, they can be significantly different due to semantic shift between domains as illustrated in Figure 2.

Here, we explain the essence of LLM. Interested readers will consult the original paper for the details. Suppose that \( T^{LM} \) is the word embeddings of the target domain induced by a language model task, and \( S^{MT} \) is the word embeddings of the source domain induced by the translation task (the embedding layer of the pre-trained model). We denote the vocabulary of \( T^{LM} \) by \( V_T \) and the vocabulary of \( S^{MT} \) by \( V_S \) and the vocabulary of words shared across both domains by \( V_{\text{shared}} = V_T \cap V_S \). Our goal is to produce embeddings \( T^{MT} \) with a vocabulary of \( V_T \) in the embedding space of \( S^{MT} \). We accomplish this by computing \( T^{MT} \) that best preserves the local topology of \( T^{LM} \) in the embedding space of \( S^{MT} \).

Concretely, for each word \( w_i \) in \( V_T \), we first take \( k \)-nearest neighbors \( N(w_i) \subset V_{\text{shared}} \) in \( T^{LM} \). We use cosine similarity as the metric for the nearest neighbor search.

Secondly, we learn local topology around \( w_i \) by reconstructing \( T_{w_i}^{LM} \) from the embeddings of its nearest neighbors as a weighted average. For this purpose, we minimize the following objective:

\[
\hat{\alpha}_i = \arg \min_{\alpha_i} \left( \| T_{w_i}^{LM} - \sum_{j \in N(w_i)} \alpha_{ij} T_{w_j}^{LM} \|^2 \right) .
\]

(1)

with constraint of \( \sum_j \alpha_{ij} = 1 \); this has an analytical solution by the method of Lagrange multipliers.

We then compute the embedding \( T_{w_i}^{MT} \) that best preserve the local topology by minimizing the following objective function:

\[
T^{MT} = \arg \min_{T^{MT}} \left( \| T_{w_i}^{MT} - \sum_{w_j \in N(w_i)} \alpha_{ij} S_{w_j}^{MT} \|^2 \right) .
\]

(2)

This optimization problem has obvious solution of

\[
T_{w_i}^{MT} = \sum_{w_j \in N(w_i)} \alpha_{ij} S_{w_j}^{MT} .
\]

(3)

Note that words shared across domains will have different embeddings after projection (\( T_{w_i}^{MT} \neq S_{w_i}^{MT} \) for \( w \in V_{\text{shared}} \)). This captures the semantic shift of words across domains. We conduct a detailed analysis of this matter in § 5.3.

4 Experimental Setup

We conduct fine-tuning with our vocabulary adaptation for distant domain adaptation on En→Ja and De→En machine translation. In what follows, we describe the experimental setup of our experiments.

4.1 Dataset and Preprocessing

In En→Ja translation, we choose Japanese-English Subtitle Corpus (JESC)\(^1\) (Pryzant et al., 2018) as the source domain and Asian Scientific Paper Excerpt Corpus (ASPEC)\(^2\) (Nakazawa et al., 2016) as

\(^1\)https://nlp.stanford.edu/projects/jesc/
\(^2\)http://orchid.kuee.kyoto-u.ac.jp/aspec/

- \( \text{En} \to \text{Ja} \)
- \( \text{JESC} \to \text{ASPEC} \)
- \# examples:
  - training (all) 2,797,388
  - development 2,000
  - testing 1,790
  - 1,812
- \# distinct words (En): 161,695 (637,377)
  - distinct words (Ja): 169,649 (384,077)
- \# shared words (En): 46,950 (7.4% in ASPEC)
  - shared words (Ja): 43,608 (11.4% in ASPEC)

| De→En | IT | Medical |
|-------|----|---------|
| \# examples
  - training (all) 337,817
  - development 2,526
  - testing 2,000 |
| \# distinct words (De): 140,508
  - distinct words (En): 70,650 (92,316)
  - shared words (De): 17,165 (12.2% in IT)
  - shared words (En): 43,608 (47.2% in IT)|

Table 1: Statistics of the source and target parallel data. # distinct/shared words are counted in the training sets.
the target domain. JESC is a parallel corpus constructed from subtitles of movies and TV shows, while ASPEC was constructed from scientific papers. These domains are substantially distant, and the ASPEC domain contains many scientific and technical terms that are unknown in the JESC domain. We follow the official splitting of training, development, and test sets, except that the last 1,000,000 sentence pairs are omitted in the training set of ASPEC corpus as they contain low-quality translations.

In De → En translation, we adopt the dataset constructed by (Koehn and Knowles, 2017) from the OPUS corpus (Tiedemann, 2012). This dataset includes multiple domains that are distant from each other and is suitable for experiments on distant domain adaptation. We choose the IT domain and the Medical domain from the dataset as the source and target domain, respectively.

Preprocessing As preprocessing for En → Ja datasets, we first tokenize the parallel data using Moses toolkit3 (v4.0) for English sentences and KyTea4 (v0.4.2) for Japanese sentences. We then truecase the English sentences by the script in the Moses toolkit following the official tutorial. As for De → En datasets, we follow the same splitting of the data, tokenization, and truecasing of (Koehn and Knowles, 2017). The statistics of the datasets are listed in Table 1.

To the tokenized datasets, we apply SentencePiece5 (v0.1.83) (Kudo and Richardson, 2018) trained from the monolingual corpus in each domain. The size of subwords is 16,000. In the training of SentencePiece, we do not concatenate the input language and output language, to maximize the portability of the pre-trained model.

From each of the preprocessed datasets, we use 1) randomly sampled 100,000 sentence-pairs or 2) all of the training set for training in the target domain, to simulate both cases that we have a small/large dataset in the target domain. We leverage the entire unaligned training set of each domain as a monolingual corpus in the domain for the reproducibility of the experiments. The monolingual corpus is used for training SentencePiece, training CBOW vectors, and data augmentation by back-translation.

4.2 Models and Embeddings
We adopt Transformer (Vaswani et al., 2017) implemented in fairseq (v0.8.0)6 (Ott et al., 2019) as the core architecture for NMT models. Major hyperparameters of the models are shown in Table 2. We evaluate the performance of the models by case-sensitive BLEU (Papineni et al., 2002). Before pre-training of models, we induce subword embeddings from the monolingual corpus using Continuous Bag-of-Words (CBOW) (Mikolov et al., 2013) to initialize the embedding layers of the NMT model.

In order to evaluate the effect of vocabulary adaptation for distant domain adaptation in NMT, we compare the following six settings (and their combinations) that use either or both the source- and target-domain parallel data.

Out-/In-domain trains a model only from the training set in the source/target domain.

Fine-tuning (FT) continues to train the Out-domain model using the training set in the target domain without any vocabulary adaptation (Luong and Manning, 2015).

Multi-domain learning (MDL) trains a model from both the source and target domain training sets. We employ domain token mixing (Britz et al., 2017) as a standard method of multi-domain learning. This method prepends a special token of the current domain (e.g., <src>) to the target sentence in training. This enforces the decoder to predict the current domain from the input, which works as regularization.

Back-translation (BT) applies a backward translation to target domain monolingual corpora in the target language (Sennrich et al., 2016a). For this back-translation, a backward Out-domain model (e.g., Ja → En) is independently trained. The subsequent fine-tuning is applied with the generated pseudo-parallel in-domain corpora and an in-domain training set.

Vocabulary Adaptation (VA) re-initializes the embedding layers7 of the Out-domain model

---

3https://github.com/dmoses-mt/mosesdecoder
4http://www.phontron.com/kytea
5https://github.com/google/sentencepiece
6https://github.com/pytorch/fairseq
7Since Transformer shares the embedding and output layers of the decoder, vocabulary adaptation is applied to the embedding layer of the encoder and the shared embedding/output layer of the decoder, respectively.
# encoder/decoder layers 6
Dim. of Transformer 2048
Init. learning rate 1e-3
Dim. of embeddings 512
Dropout rate 0.1
Vocab. size (encoder) 16k
Beam size for decoding 5
Vocab. size (decoder) 16k

to embeddings projected from the target domain by cross-domain embedding projection prior to fine-tuning. We compare two methods of cross-domain embedding projection: the linear orthogonal transformation (Linear, § 3.1) and the LLM (§ 3.2). For LLM, the size of nearest neighbors, k, is fixed to 10 (default).

Among the above models, In-domain and Out-domain do not perform domain adaptation. FT, MDL, and BT are baseline domain adaptation models in which the vocabulary mismatch problems can occur.

Note that only MDL assumes that the target domain is given before training with the source domain corpus. In the setting, we jointly use the source-domain and target-domain training sets for training subword tokenization models, CBOW vectors, and training NMT models. To choose the best model, we use the target-domain development set.

We trained each model using Adam optimizer (Kingma and Ba, 2015). During training, the learning rate is linearly increased for warm-up in the first 4k training steps, and then decayed proportionally to the inverse square root of the number of updates. Prior to fine-tuning, we reset the optimizer and the learning rate, and then continue training on the training set in the target domain. For vocabulary adaptation, we replace the embedding layers of the pre-trained models by the word embeddings projected from the target domain except for the four special tokens (<pad>, <unk>, </s>, <s>).

## Results and Analysis

### 5.1 Main Results

Table 3 and Table 4 show the main results on the distant domain adaptations. First, the low BLEU scores of Out-domain show how the domain mismatch degrades the performance of models, as pointed out in (Koehn and Knowles, 2017). The proposed method with locally-linear mapping, VA (LLM) largely outperformed FT and MDL, and achieved better or comparable results to BT. Surprisingly, VA (LLM) also achieved slightly better BLEU scores than FT (+0.87 pt on En → Ja and +0.98pt on De → En), even when all of the target-domain training set was available and thus the models did not obtain additional supervision from the monolingual corpus.

As all of the models employed subword tokenization, there are no out-of-vocabulary tokens in the corpora. The results imply the large impact of employing the subword tokenization and embeddings which were tailored to each domain. Meanwhile, the BLEU scores of VA (Linear) were quite lower than those of VA (LLM), although the two models were based on the same pre-trained models and data. These results highlight the superiority of LLM over linear transformation in cross-domain/task embedding projection.

We also evaluated the combination of BT and VA (LLM) to confirm their orthogonality, since...
both models exploit target-domain monolingual corpora for domain adaptation. The only difference in the setting from VA (LLM) is that we simply add the data generated by BT to the target-domain training set. The results of BT + VA (LLM) show that employing vocabulary adaptation largely improved the BLEU score (+3.92 pt from BT) in the En → Ja translation when the in-domain corpus was small (100k). In the De → En translation, back-translation improved neither FT nor VA (LLM).

Overall, we conclude that these results confirm the need for vocabulary adaptation and orthogonality to back-translation in distant domain adaptation, mainly when the size of the target-domain parallel corpus is relatively small.

**Translation example**  Table 5 shows translation examples in the En→Ja translation. In the examples, the underlined words in the reference ‘押出’ and ‘ポリマ アロイ’ correspond to ‘extrusion’ and ‘polymer alloys’, respectively. Although the words are out of vocabulary in the source domain, BT + VA (LLM) successfully translated them.

In other examples, we observed that the outputs generated by BT + VA contained various target-domain-specific words. To quantitatively examine that, we calculate the percentage of distinct words included in both of the generated outputs and the references. The outputs generated by BT + VA (LLM), BT, and FT contained 57.1 %, 54.0 %, and 49.8 % of distinct words in the references, respectively.

**5.2 Training Steps**

Table 6 shows the number of updates until convergence in En → Ja translation with the 100k training set and 2000k monolingual corpus in the target domain.

| Model         | # Updates |
|---------------|-----------|
|               | Pre-training | Fine-tuning |
| No adaptation |            |             |
| Out-domain    | 28,750     | -           |
| In-domain     | 4,047      | -           |
| Baselines     |            |             |
| FT            | 28,750     | 2,160       |
| BT            | 56,350     | 246,510     |
| MDL           | 21,080     | -           |
| Proposed      |            |             |
| VA (Linear)   | 28,750     | 1,988       |
| VA (LLM)      | 28,750     | 2,556       |
| BT + VA (LLM) | 56,350     | 91,212      |

Table 6: The number of updates until convergence in En → Ja translation with the 100k training set and 2000k monolingual corpus in the target domain.
Nearest neighbors in ASPEC-CBOW embedding space

- **page**
- **browser**, **documents**, **server**, **book**, **menu**
- **experimentally**, **theoretically**, **systematically**, **by**
- **experiments**, **experimental**

Nearest neighbors in JESC-NMT embedding space

via linear transformation (Linear)

- **page**
- **text**, **journal**, **messages**, **magazine**, **mail**
- **experimentally**, **waves**, **ve**, **we**, **they**, **em**

via locally-linear mapping (LLM)

- **book**, **server**, **mail**, **documents**, **newspaper**
- **experimentally**, **experiment**, **theoretically**, **by**
- **experiments**, **experimental**

Table 7: Top-5 nearest neighbors of ‘page’ and ‘experimentally’ in the ASPEC-CBOW embedding space and JESC-NMT embedding space via cross-domain embedding projection: **bold-faced** subwords are the nearest neighbors shared across both top-5.

Table 7 shows the nearest neighbors of the words ‘page’ and ‘experimentally’ in both ASPEC-CBOW and JESC-NMT embedding spaces. The nearest neighbors are listed for each word, with the ASPEC-CBOW neighbors on top and the JESC-NMT neighbors below. The nearest neighbors are bolded for those that are shared across both domains.

5.3 Qualitative Analysis of Cross-domain Embeddings

The advantage of our approach is that it not only adjusts the vocabulary (set of subwords) to the target domain, but also adjusts the meanings of subwords (embeddings) to the target domain. In this section, we examine to what extent our vocabulary adaptation captures the semantic shift across domains. Here, we focus on the nearest neighbors of a subword embedding in the target domain (ASPEC) space and the source domain (JESC) space. In vocabulary adaptation, the nearest neighbors of a subword embedding should be unchanged even after mapping and compared in the source domain, to keep the meaning in the target domain.

In Table 7, the upper rows show the ground truth nearest neighbors of each word. To obtain the nearest neighbors, we compare the CBOW vectors of the two words trained in ASPEC, with the same CBOW vectors of other words by cosine similarity. On the other hand, the lower rows show the cross-domain nearest neighbors. To obtain these nearest neighbors, we compare the cross-domain embeddings mapped by each method with the NMT model’s embeddings trained by all parallel data of JESC. We raise two words as examples: ‘page’ which appears in both domains, and ‘experimentally’ which is only in the ASPEC domain.

‘page’ is a subword that can have different meanings across domains. While the CBOW vector and the embedding projected by LLM have a meaning of ‘server’ (i.e., web page), the embedding projected by linear transformation lost it.

‘experimentally’ is a subword that the target-domain (ASPEC) vocabulary only contains. As shown in Figure 2, mapping of target-domain-specific subword embeddings is likely to fail due to the difference of topology in embedding space. We found that LLM accurately induced its embedding in the embedding space of the trained model while linear transformation failed, and the nearest neighbors were irrelevant.

These observations indicate the capability of LLM in cross-task/domain embedding projection.

6 Conclusions

In this study, we have tackled the crux of the vocabulary mismatch problems in domain adaptation for NMT, and proposed vocabulary adaptation, a simple but effective method for adapting the vocabulary of pre-trained NMT model to the target domain, to perform effective fine-tuning. Regarding domains as independent languages/tasks, our method makes wide-coverage word embeddings induced from target-domain monolingual data to be compatible with the model pre-trained from the source-domain data.

We explored two methods to project word embeddings across two domains: linear transformation and LLM. The experimental results on English to Japanese translation and on English to German translation confirm that our domain adaptation method with the locally-linear embedding projection (LLM) dramatically improved the translation performance.

The best-performing vocabulary adaptation, LLM, is simple and easy to reproduce thanks to the analytical solution. It is also applicable to a wider-range of text-generation models other than NMT models, and can be combined with existing fine-tuning-based domain adaptation. We will release all the codes to promote the reproducibility of our results.
References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proceedings of the third International Conference on Learning Representations (ICLR 2015).

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), pages 1538–1548.

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 2017), pages 118–126.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (ACL 2017), pages 385–391.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. CoRR, abs/1612.06897.

Sluhao Gu, Yang Feng, and Qin Liu. 2019. Improving domain adaptation translation with domain invariant and specific information. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019), pages 3081–3091.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Hui-E-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. CoRR, abs/1503.03535.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015), pages 1–10.

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In Proceedings of the 2nd Workshop on Neural Machine Translation and Generation (WNMT 2018), pages 36–44.

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

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 2017), pages 372–378.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In Proceedings of the First Workshop on Neural Machine Translation (WNMT 2017), pages 28–39.

Taku Kudo. 2018. Subword regularisation: Improving neural network translation models with multiple subword candidates. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), pages 66–75.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2018), pages 66–71.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W. Black. 2015. Character-based neural machine translation. CoRR, abs/1511.04586.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2015), pages 76–79.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), pages 1054–1063.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015), pages 11–19.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26 (NIPS 2013), pages 3111–3119.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), pages 2204–2208.
Masato Neishi, Jin Sakuma, Satoshi Tohda, Shunosuke Ishiwatari, Naoki Yoshinaga, and Masashi Toyoda. 2017. A bag of useful tricks for practical neural machine translation: Embedding layer initialization and large batch size. In Proceedings of the 4th Workshop on Asian Translation (WAT 2017), pages 99–109.

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 2019), pages 48–53.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), pages 311–318.

Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. 2018. JESC: Japanese-English subtitle corpus. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Jin Sakuma and Naoki Yoshinaga. 2019. Multilingual model using cross-task embedding projection. In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL 2019), pages 22–32.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), pages 86–96.

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

Brian Thompson, Huda Khayrallah, Antonios Anastasopoulos, Arya D. McCarthy, Kevin Duh, Rebecca Marvin, Paul McNamee, Jeremy Gwinnup, Tim Anderson, and Philipp Koehn. 2018. Freezing subnetworks to analyze domain adaptation in neural machine translation. In Proceedings of the Third Conference on Machine Translation: Research Papers (WMT 2018), pages 124–132.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems 30 (NIPS 2017), pages 5998–6008.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pages 1482–1488.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2015), pages 1006–1011.