Progressive Translation: Improving Domain Robustness of Neural Machine Translation with Intermediate Sequences

Chaojun Wang1  Yang Liu2  Wai Lam1

1The Chinese University of Hong Kong
2Microsoft Cognitive Services Research
cj.wang@link.cuhk.edu.hk

Abstract

Previous studies show that intermediate supervision signals benefit various Natural Language Processing tasks. However, it is not clear whether there exist intermediate signals that benefit Neural Machine Translation (NMT). Borrowing techniques from Statistical Machine Translation, we propose intermediate signals which are intermediate sequences from the "source-like" structure to the "target-like" structure. Such intermediate sequences introduce an inductive bias that reflects a domain-agnostic principle of translation, which reduces spurious correlations that are harmful to out-of-domain generalisation. Furthermore, we introduce a full-permutation multi-task learning to alleviate the spurious causal relations from intermediate sequences to the target, which results from exposure bias. The Minimum Bayes Risk decoding algorithm is used to pick the best candidate translation from all permutations to further improve the performance. Experiments show that the introduced intermediate signals can effectively improve the domain robustness of NMT and reduces the amount of hallucinations on out-of-domain translation. Further analysis shows that our methods are especially promising in low-resource scenarios.

1 Introduction

A spectrum of studies recently arose in Natural Language Processing (NLP), which incorporates intermediate supervision signals into the model by simply converting the intermediate signals into textual sequences and prepending or appending these sequences to the output sequence. It benefits tasks such as math word problems (Wei et al., 2022), commonsense reasoning (Liu et al., 2022), programs execution (Nye et al., 2022), summarisation (Narayan et al., 2021), etc. This trend further

The work described in this paper is substantially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200620).

Figure 1: An illustration of the transformation from a source sentence to the target translation and its analogy with vision. src: source; tgt: target; lex: word-by-word translation; ali: reorders lex monotonically based on word alignments.

das ist sie. That is she. There she is.

triggered the collection of a new dataset with intermediate results (Lewkowycz et al., 2022) and corresponding theoretical analysis (Wies et al., 2022). Intermediate supervision signals show consistent benefits to these various sequence generation tasks and Neural Machine Translation (NMT) is a basic and typical sequence generation task in the NLP community. However, it remains an open question whether and how intermediate signals can be defined and leveraged for NMT.

Meanwhile, previous studies (Koehn and Knowles, 2017; Müller et al., 2020) found that NMT suffers from poor domain robustness, i.e. the generalisation ability to unseen domains. Such an ability not only has theoretical meaning, but also has practical value since: 1) the target domain(s) may be unknown when a system is built; 2) some language pairs may only have training data for limited domains. Since the recent study (Wei et al., 2022) in intermediate supervision signals showed a benefit of such signals on out-of-domain generalisation, we expect intermediate signals may benefit domain robustness in NMT.

Different from math problem-solving tasks, machine translation tasks do not have explicit intermediate results to serve as the intermediate signals. A recent work (Voita et al., 2021b) found that NMT acquires the three core SMT competencies, target-side language modelling, lexical translation and reordering in order during the course of the training. Inspired by this work, we borrow tech-
niques in SMT to produce intermediate sequences as the intermediate signals for NMT. Specifically, we first obtain the word alignments for the parallel corpus and use it to produce the word-for-word translations (lex) and the aligned word-for-word translations (ali) to resemble the lexical translation and reordering competencies in SMT. As shown in Figure 1, the intermediate sequences resemble structurally approaching the target from the source progressively, which shares a similar spirit of how humans do translation or reasoning about translation step by step, thus named Progressive Translation.

Our intuition is that these intermediate sequences inject an inductive bias about a domain-agnostic principle of the transformation between two languages, i.e. word-for-word mapping, then reordering, and finally refinement. Such a bias limits the learning flexibility of the model but prevents the model from building up some spurious correlations (Arjovsky et al., 2019) which harm out-of-domain performance.

However, previous works have shown that NMT is prone to overly relying on the target history (Wang and Sennrich, 2020; Voita et al., 2021a), which is partially correlated with exposure bias (Ranzato et al., 2016) (a mismatch between training and inference), especially under domain-shift. Simply prepending these introduced intermediate sequences to the target would introduce spurious causal relationships from the intermediate sequences to the target. As a result, these intermediate sequences would potentially mislead the model about the prediction of the target, due to erroneous intermediate sequences during inference. To alleviate this spurious causal relationship, we introduce the full-permutation multi-task learning framework, where the target and intermediate sequences are not limited to literally understandable translations. As shown in Figure 1, the intermediate sequences resemble the deliberation process of humans. Similar to Narayan et al. (2021), Progressive Translation (PT) augments data for the whole training set and the intermediate sequences are not limited to literally understandable translations. Similar to Wei et al. (2022), sequences augmented by PT resemble approaching the output from the input.

Intermediate Supervision Signals. Some existing works in the broader NLP community try to incorporate intermediate sequences into the model. We take two typical examples of them to better distinguish our work from other works. Narayan et al. (2021) uses an entity chain as the intermediate sequence for summarisation. Wei et al. (2022) produces intermediate sequences resembling the deliberation process of humans. Similar to Narayan et al. (2021), Progressive Translation (PT) augments data for the whole training set and the intermediate sequences are not limited to literally understandable sequences. Similar to Wei et al. (2022), sequences augmented by PT resemble approaching the output from the input.

Data Augmentation of Domain Robustness in NMT. Existing works in data augmentation try to improve the domain robustness of NMT by introducing more diverse synthetic training examples (Ng et al., 2020) or auxiliary tasks where the target history is less informative (Sánchez-Cartagena et al., 2021) named MTL-DA framework. The main difference between our PT framework and the MTL-DA framework is that the MTL-DA framework treats each target-side sequence as an independent task conditioned on the source, whereas PT also encourages the model to learn the transformational relations between any pair of target-side sequences, which may help the model to generalise better across domains.
Multi-task Learning:

<123> is a control token indicating the order of three sequences. 1: lex; 2: ali; 3: tgt, then <123> is for the task where the target is in order of lex, ali and tgt. <lex>, <ali>, <tgt> is the special tokens prepended to lex, ali, tgt separately.

Data Augmentation:

Old training pair:
Source: Das ist sie.
Target: There she is.

New training pairs:
Source: <123> Das ist sie.
Target: <lex> That is she. <ali> That she is. <tgt> There she is.
Source: <321> Das ist sie.
Target: <tgt> There she is. <ali> That she is. <lex> That is she.

... six (3!) permutations in total

Figure 2: An illustration of the proposed intermediate sequences and multi-task learning framework. src: source.


Statistical Machine Translation in NMT. The intermediate sequences of PT are produced using the word alignments and reordering components in Statistical Machine Translation (SMT). There are works on improving NMT with SMT features and techniques (He et al., 2016; Chen et al., 2016; Du and Way, 2017; Zhao et al., 2018). However, these works either modify the architecture of the neural network or require more than one model to produce the translation (e.g. a rule-based pre-ordering model and a NMT model etc.). To the best of our knowledge, we are the first to incorporate features from SMT into NMT by converting the features into textual sequences and prepending these to the target without requiring extra models or modifying the neural architecture.

3 Approach

3.1 Intermediate Sequences

The traditional SMT decomposes the translation task into distinct components where some features could potentially be the intermediate supervision signals. More recently, Voita et al. (2021b) found that NMT acquires the three core SMT competencies, i.e. target-side language modelling, lexical translation and reordering, in order during the course of training. Inspired by this work, we produce word-for-word translations and aligned word-for-word translations as the intermediate sequences to resemble the lexical translation and reordering components separately using the word alignments component in SMT.

As shown in Figure 2 Data Augmentation part, for each source-target parallel sequence in the training corpus, we augment their target sequences with two extra intermediate sequences, lex and ali. The two intermediate sequences are prepended to the target to form an augmented target.

lex: The source sequence is word-for-word translated based on a bilingual lexicon obtained from the parallel training corpus. Tokens that are not in the lexicon are copied into lex.

ali: lex is reordered so that the word alignments from the target to lex is monotonic. The word alignments used here are target-to-source alignments because it is equivalent to the target-to-lex alignments since lex is word-for-word mapped from the source. The words in the target which is assigned to "NULL" are omitted during reordering.

lex, ali and target (tgt) are prefixed with a special token separately for extracting the corresponding sequence from the predicted output. The one-to-many (both source-to-target and target-to-source) word alignments are obtained with mgiza++ (Gao and Vogel, 2008; Och and Ney, 2003)\(^1\), a SMT word alignments tool, on the in-domain training corpus, following the default parameter provided in train-model.perl by Moses (Koehn et al., 2007)\(^2\). The one-to-one word alignments are built by computing the intersection between the one-to-many word alignments in both directions. The bilingual lexicon is obtained by associating each source word

\(^1\)https://github.com/moses-smt/mgiza
\(^2\)https://github.com/moses-smt/mosesdecoder/blob/master/scripts/training/train-model.perl
Figure 3: Causal graphs for the source and three target-side sequences. Solid arrow denotes casual dependence and dashed arrow represents the statistical correlation between two variables. Left: relations if we simply prepend lex and ali to the target. Right: relations after full-permutation multi-task learning.

to the target word it is most frequently aligned within the one-to-one word alignments.

The learning of word alignments and transformations of lex and ali are at the word level. The BPE (Sennrich et al., 2016) word segmentation is trained on src-tgt parallel data as normal and applied to both source-target parallel sequences and intermediate sequences (the target-language vocabulary is applied to split the words in the intermediate sequences).

We expect that the introduced intermediate sequences would benefit the domain robustness of NMT. Because the proposed intermediate sequences serve as a supervision signal to provide the model with an explicit path for learning the transformational relations from source to target. Such signals inject an inductive bias about one kind of domain-agnostic principle of the transformation between two languages, i.e. word-for-word mapping, then reordering, finally refinement. This injected bias limits the learning flexibility of the neural model but prevents the model from building up some spurious correlations which harm out-of-domain performance.

3.2 Spurious Causality Relationship

To introduce these intermediate sequences as intermediate supervision signals to the model, we prepend them to the output sequence in training. However, simply prepending these produced intermediate sequences to the target would potentially introduce spurious causality relationships from pre-sequence to post-sequence. For example, prepending lex, ali to the target would introduce the causal relationships of lex → ali → tgt. These are spurious causality relationships because the model is highly unlikely to get the gold-standard pre-sequences (lex or ali) as in the training during inference, especially under the domain-shift where the performance is relatively poor. Therefore, the model should learn that source (input) is the only reliable information for any target-side sequences. Note that such spurious causality relationship in principle results from a mismatch between training and inference of the standard training-inference paradigm of NMT, which is termed exposure bias by the community.

Intuitively, if the model could predict the target-side sequences in any order, then the causality relationship between target-side sequences should be reduced. Therefore, we propose to fully permute the target-side sequences, i.e. intermediate sequences (lex or ali) and the target sequence (tgt). Figure 2 illustrates the training data after permutation when we prepend both lex and ali to the target. The source is prefixed with a control token for each permutation, i.e. 1: lex; 2: ali; 3: tgt, then <123> is the control token for the permutation where the target is in the order of lex, ali and tgt.

As shown in Figure 3, with the permutation, we create counterfactual data which disentangles the causal relations of lex → ali → tgt and enhances the causal relations from source to each of these three sequences. Therefore, the full-permutation multi-task training better balances the model’s reliance on the source and target history, at least on pre-sequence(s).

3.3 Minimum Bayes Risk Decoding

From our preliminary experiments, we found that various test sets prefer different generation orders of the permutation. For example, order lex-ali-tgt performs best on some test sets whereas tgt-ali-lex performs best on some other test sets. Therefore, we suspect that the translation quality would be further improved if we could dynamically select the best candidate translations from all permutations. Inspired by (Eikema and Aziz, 2021), we use Minimum Bayes Risk (MBR) decoding to select a consensus translation from all permutations.

MBR aims to find a translation that maximises expected utility (or minimises expected risk) over the posterior distribution. In practice, the posterior distribution is approximated by drawing a pool of samples \( \mathcal{S} = (s_1, \ldots, s_n) \) of size \( n \) from the model:

\[
y^* = \arg\max_{s_i \in \mathcal{S}} \frac{1}{n} \sum_{i=1}^{n} u(s_i, s_j) \tag{1}
\]

where \( u \) is the utility function to compute the similarity between two sequences. In our experiment,
the samples $S$ are translations from all permutations.

Following Eikema and Aziz (2021), we use BEER (Stanojević and Sima'an, 2014) as the utility function, and the released toolkit\(^1\) for MBR decoding.

4 Experiments

4.1 Dataset

We work on three datasets involving two language pairs, which were used in previous works on the domain robustness in NMT (Sánchez-Cartagena et al., 2021; Ng et al., 2020).

IWSLT'14 DE→EN IWSLT'14 (Cettolo et al., 2014) German→English (DE→EN) is a commonly used small-scale dataset in NMT, which consists of 180,000 sentence pairs in the TED talk domain. Following Sánchez-Cartagena et al. (2021), the validation and in-domain (ID) testing sets are tst2013 and tst2014 separately; and out-of-domain (OOD) test sets consist of IT, law and medical domains from OPUS (Lison and Tiedemann, 2016) collected by Müller et al. (2020)\(^4\).

OPUS DE→EN & Allegra DE→RM are two benchmarks of domain-robustness NMT released by Müller et al. (2020). OPUS comprises five domains: medical, IT, law, koran and subtitles. Following Ng et al. (2020), we use medical as ID for training (which consists of 600,000 parallel sentences) and validation and the rest of four domains as OOD test sets. Allegra (Scherrer and Cartoni, 2012) German→Romansh (DE→RM) has 100,000 sentence pairs in law domain. The test OOD domain is blogs, using data from Convivenza.

We tokenise and truecase all datasets with Moses and use shared BPE with 10,000 (on IWSLT'14) and 32,000 (on OPUS and Allegra) for word segmentation (Sennrich et al., 2016).

4.2 Models and Evaluation

All experiments are done with the Nematus toolkit (Sennrich et al., 2017) based on the Transformer architecture (Vaswani et al., 2017)\(^5\). The baseline is trained on the training corpus without using intermediate sequences. We follow Wang and Sennrich (2020) to set hyperparameters (see Appendix) on three datasets. For our framework, we scale up the token batch size proportional to the length of the target for a fair comparison, e.g. if the target-side sequence is three times longer than the original target, we scale up the batch size to three times as well.\(^6\) The performance of the original order (lex)-(ali)-tgt is used for validation and testing. We conduct early-stopping if the validation performance underperforms the best one over 10 times of validation in both the translation quality (BLEU) and the cross entropy loss.

We also compare to two recently proposed methods of domain robustness in NMT. SSMBA (Ng et al., 2020) generates synthetic training data by moving randomly on a data manifold with a pair of corruption and reconstruction functions. Reverse+Mono+Replace (Sánchez-Cartagena et al., 2021) (RMP) introduces three auxiliary tasks where the target history is less informative.

We report cased, detokenised BLEU (Papineni et al., 2002) with SacreBLEU (Post, 2018)\(^7\). Each experiment is independently run for three times, and we report the average and standard deviation to account for optimiser instability.

4.3 Results

We test our proposal mainly on IWSLT'14 DE→EN. Table 1 summarises the results. \(^1\) is the baseline system which is trained on parallel corpus only without any data augmentation. The average OOD is computed by averaging results across all OOD test sets.

**Single lex benefits OOD whereas ali does not.** Firstly, we simply prepend the produced intermediate sequence(s) (any one of them and both of them in the order of lex-ali) to the target sequence. Results show that single lex (\(^2\)) significantly improves the OOD performance by 2.2 BLEU, at the cost of 0.9 BLEU decrease in in-domain performance. However, the introduction of ali deteriorates the performance on both in-domain (ID) and OOD test sets (\(^3\) and \(^4\)). We argue that this comes from the reason that the learning of generating ali is more difficult than generating lex (ali needs an extra reordering step and also the produced ali is noisy due to the word alignment errors). As a result, ali is more erroneous than lex during inference. Therefore, the generation quality of the

---

\(^1\)https://github.com/Roxot/mbr-nmt

\(^2\)https://github.com/ZurichNLP/domain-robustness

\(^3\)https://github.com/chaojun-wang/progressive-translation

\(^4\)Scaling up the token batch size only brings negligible improvement on the baseline.

\(^7\)Signature: BLEUI#:1lc:mixedle:noltok:13als:expiv:2.1.0
target deteriorates due to its causal dependency on ali.

**ali benefits OOD with the support of permutation multi-task learning.** We try to alleviate the problem by introducing the permutation multi-task learning on top of (2)→(4). Results show that the permutation successfully alleviates the deterioration of introducing ali, bringing positive results for both ID and OOD (3)→(6), (4)→(7). With the permutation, a single ali intermediate sequence (6) can improve OOD over the baseline by 2 BLEU and the combination of lex and ali (7) bring further improvement on OOD over single lex (2) or single ali (6) by 0.5 and 0.7 BLEU respectively. The permutation shows a negative effect on single lex (2)→(5). Because the lex is very easy to learn, few error would occur when predicting lex. Therefore, permutation is not effective and even has negative effects as it makes the neural model hard to focus on learning the task of lex-tgt, leading to inferior performance.

**MBR decoding brings further improvement.** For the lex, ali, tgt with permutation, there are six permutations in total. We dynamically select a consensus translation over each input data by performing MBR decoding over translation from all permutations. Results show MBR (7)→(8) could further improve the OOD and ID performances by 0.4 and 0.6 BLEU respectively, and outperforms baseline OOD by 3.1 BLEU at the cost of 1.6 BLEU decrease in ID.

**Results on other datasets and comparison with existing methods.** As (8) achieves the highest OOD performance and (2) achieves relatively high OOD and ID performance with simpler techniques, we name (8) as PT$_{full}$ and (2) as PT$_{simple}$ and evaluate these two methods on another two domain-robustness datasets (OPUS DE→EN and Allegra DE→RM). Table 2 lists the results.

Table 1: Average BLEU (↑) and standard deviation of ablation results on in-domain and out-of-domain test sets on IWSLT’14 DE→EN. permu: permutation.

<table>
<thead>
<tr>
<th>ID</th>
<th>Augmentation</th>
<th>In-Domain</th>
<th>IT</th>
<th>Law</th>
<th>Medical</th>
<th>average OOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transformer</td>
<td>32.1±0.38</td>
<td>14.7±0.21</td>
<td>10.1±0.38</td>
<td>17.0±0.25</td>
<td>13.9±0.19</td>
</tr>
<tr>
<td>2</td>
<td>lex+tgt</td>
<td>31.2±0.50</td>
<td>16.6±0.26</td>
<td>11.1±0.23</td>
<td>20.7±0.66</td>
<td>16.1±0.30</td>
</tr>
<tr>
<td>3</td>
<td>ali+tgt</td>
<td>25.8±1.57</td>
<td>14.4±2.54</td>
<td>4.5±6.00</td>
<td>17.9±1.32</td>
<td>12.2±2.25</td>
</tr>
<tr>
<td>4</td>
<td>lex+ali+tgt</td>
<td>25.5±1.72</td>
<td>9.4±1.14</td>
<td>3.1±2.31</td>
<td>11.3±6.70</td>
<td>7.9±1.71</td>
</tr>
<tr>
<td>5</td>
<td>2 + permu</td>
<td>30.1±1.55</td>
<td>15.5±0.50</td>
<td>7.2±5.48</td>
<td>19.0±1.08</td>
<td>13.9±2.18</td>
</tr>
<tr>
<td>6</td>
<td>3 + permu</td>
<td>30.6±0.30</td>
<td>16.9±1.00</td>
<td>10.8±0.40</td>
<td>19.9±0.60</td>
<td>15.9±0.53</td>
</tr>
<tr>
<td>7</td>
<td>4 + permu</td>
<td>29.9±0.32</td>
<td>18.2±0.89</td>
<td>10.8±0.10</td>
<td>20.7±0.40</td>
<td>16.6±0.37</td>
</tr>
<tr>
<td>8</td>
<td>7 + MBR</td>
<td>30.5±0.21</td>
<td>17.7±0.72</td>
<td>11.8±0.1</td>
<td>21.6±0.49</td>
<td>17.0±0.35</td>
</tr>
</tbody>
</table>

Baselines (Transformer) in cited works (RMP and SSMA) are trained under inappropriate hyperparameters, e.g. on IWSLT’14, the cited works uses default hyperparameters for the WMT dataset (more than 10 times larger than IWSLT’14). To enable better comparison by other researchers, we train the Transformer with the appropriate hyperparameters provided by Wang and Sennrich (2020) to build strong baselines, which outperform those in the cited works. We re-implement the other two DA methods based on our baseline for comparison.

Results show that both PT$_{simple}$ and PT$_{full}$ perform most effectively on IWSLT’14 OOD, surpassing the existing methods by 0.7-2.3 BLEU. On the other two new datasets, PT$_{simple}$ and PT$_{full}$ show consistent OOD improvement, outperforming our baseline (Transformer) by 1.1-1.6 BLEU and 1.1-1.2 BLEU on OPUS and DE→RM dataset respectively. The ID performance of PT$_{simple}$ and PT$_{full}$ on these two datasets is less affected than on IWSLT’14, at the cost of 0.3-0.4 BLUE decrease on OPUS and even no decrease on the Allegra DE→RM.

PT$_{full}$ significantly outperforms PT$_{simple}$ OOD on OPUS DE→EN and they show negligible ID differences. For Allegra DE→RM, PT$_{simple}$ and PT$_{full}$ shows similar OOD and ID performance.

5 Analysis

BLEU score indicates that the proposed methods can improve domain robustness. In this section, we investigate the reduction of hallucinations and performance on larger datasets of our methods.

5.1 Hallucinations

Hallucinations are more pronounced in out-of-domain translation, and their misleading nature makes them particularly problematic. Therefore, many works have been conducted on hallucinations,
involving detection of hallucinations (Zhou et al., 2021; Guerreiro et al., 2022; Dale et al., 2022), exploration of the causes of hallucinations (Raunak et al., 2021; Yan et al., 2022), and finding solutions for hallucinations (Miao et al., 2021; Müller and Sennrich, 2021) etc.

To test our methods for reducing the hallucinations under domain shift, we manually evaluate the proportion of hallucinations on IWSLT’14 and OPUS (DE→EN) OOD test sets. We follow the definition and evaluation by Müller et al. (2020), considering a translation as a hallucination if it is (partially) fluent and its content is not related to the source (inadequate). We report the proportion of such hallucinations in each system.

The manual evaluation is performed by two students who have completed an English-medium university program. We collect ~3000 annotations for 10 configurations. We ask annotators to evaluate translations according to fluency and adequacy. For fluency, the annotator classifies a translation as fluent, partially fluent or not fluent; for adequacy, as adequate, partially adequate or inadequate. We report the inter-annotator and intra-annotator agreement (N=150) of manual evaluation.

Table 3: Inter-annotator (N=300) and intra-annotator agreement (N=150) of manual evaluation.

<table>
<thead>
<tr>
<th>annotation</th>
<th>inter-annotator</th>
<th>intra-annotator</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P(A)</td>
<td>P(E)</td>
</tr>
<tr>
<td>fluency</td>
<td>0.52</td>
<td>0.31</td>
</tr>
<tr>
<td>adequacy</td>
<td>0.68</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Table 4: Proportion of hallucinations (↓) and BLEU (↑) on out-of-domain test sets over IWSLT’14 and OPUS (DE→EN).

<table>
<thead>
<tr>
<th>Augmentation</th>
<th>IWSLT’14</th>
<th>OPUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer</td>
<td>11% (13.9)</td>
<td>39% (11.0)</td>
</tr>
<tr>
<td>RMP</td>
<td>9% (14.7)</td>
<td>30% (12.6)</td>
</tr>
<tr>
<td>SSMBA</td>
<td>6% (15.4)</td>
<td>28% (12.1)</td>
</tr>
<tr>
<td>PTsimple</td>
<td>5% (16.1)</td>
<td>28% (12.1)</td>
</tr>
<tr>
<td>PTfull</td>
<td>7% (17.0)</td>
<td>30% (12.6)</td>
</tr>
</tbody>
</table>

Table 4: Proportion of hallucinations and BLEU.

5.2 Tendency by scaling up the corpus size
Since the size of the training corpus in the previous experiments ranges from 0.1M to 0.6M (million) samples, which is a low-resource setting for NMT, here we investigate the performance of our methods when scaling up the corpus size. We use subtitles domain from OPUS as the in-domain training data (because it has around 20M sentence pairs) and the rest four domains as the OOD test sets. We use the first 0.2M, 2M and 20M samples in the corpus as the training data separately. We follow

9431
the same data preprocessing as for OPUS (medical). The hyperparameters for training the model are the same as those for IWSLT’14 when the corpus size is 0.2M and those for OPUS (medical) when the corpus size is 2M. For the corpus size of 20M, we increase the token batch size to 16384 instead of 4096 and keep the rest of the hyperparameters the same as for the 2M corpus size. Similarly, each experiment is independently run for three times and we report the average result.

Results are shown in Figure 4. As expected, increasing the corpus size (0.2M-20M) improves both ID and OOD performance for all systems. When the corpus size is small (0.2M), PT\textsubscript{full} (red line) shows a considerable improvement in OOD over the baseline (blue line) by 4.3 BLEU and even slightly benefits ID, surpassing the baseline by around 0.9 BLEU. However, scaling up the corpus size (0.2M-20M) narrows the gap of OOD improvement (4.3-0.9 BLEU) between the baseline and PT\textsubscript{full}, and widens the ID deterioration from +0.9 to -1.6 BLEU.

In general, PT\textsubscript{simple} (green line) follows a similar tendency as PT\textsubscript{full}, compared to the baseline. However, PT\textsubscript{simple} underperforms the baseline at the corpus size of 2M. By a close inspection, we found that the training of PT\textsubscript{simple} is relatively unstable. The standard deviations of PT\textsubscript{simple} for OOD are 1.38, 2.49 and 0.24 on 0.2M, 2M and 20M corpus size respectively, whereas the standard deviations of PT\textsubscript{full} are 0.47, 0.27 and 0.52 respectively. This indicates that the training of PT\textsubscript{simple} is less stable than PT\textsubscript{full} when the corpus size is 0.2M-2M. The better stability of PT\textsubscript{full} may come from its permutation multi-task learning mechanism.

PT\textsubscript{simple} always underperforms PT\textsubscript{full} on OOD for any corpus size. PT\textsubscript{simple} shows slightly better ID performance than PT\textsubscript{full} when the corpus size is large (2M-20M) but underperforms PT\textsubscript{full} on ID performance in low resource setting where the corpus size is 0.2M.

6 Conclusion

Our results show that our introduced intermediate signals effectively improve the OOD performance of NMT. Intermediate sequence \textit{lex} can benefit OOD by simply prepending it to the target. \textit{ali} is more likely to be erroneous during inference than \textit{lex}, which results in degenerated target due to the spurious causal relationship. Our proposed permutation multi-task learning successfully alleviates the problem and manifests the effectiveness of \textit{ali}. Experiments also confirm that the MBR algorithm can further improve the performance by dynamically selecting a consensus translation from all permutations. The human evaluation shows that the proposed methods substantially reduce the number of hallucinations of the out-of-domain translation. Experiments on the larger corpus sizes indicate that our methods are especially promising in the low-resource scenarios.

Our work is the first attempt to complete the puzzle of the study of intermediate signals in NMT, and two new ideas may benefit this study in other areas: 1) thinking intermediate signals from the intermediate structures between the transformation from the input to the output; 2) the permutation multi-task learning, instead of only pre/appending intermediate sequences to the output sequence. The permutation multi-task learning + MBR decoding framework is also a potential solution for any multi-pass generation tasks (e.g. speech translation), which suffer from the error propagation problem. The problem is alleviated with the permutation which
disentangles causal relations between intermediate and final results. Finally, our work provides a new perspective of data augmentation in NMT, i.e. augmenting data by introducing extra sequences instead of directly modifying the source or target.

Acknowledgements

We thank Tianzi Wang for insightful suggestions and pointers to Minimum Bayes Risk decoding algorithm and Zengrui Jin for computing resources.

7 Limitations

The way we use the intermediate sequences is to concatenate new sequences and the target sequence as the new target. As a result, the length of the target increases linearly with the number of intermediate sequences introduced, which increases the cost of inference. In the meantime, Minimum Bayes Risk decoding needs to do prediction multiple times under different control tasks, which further increases the computational cost. However, there are potential solutions to compromise between the computational cost and quality, e.g. learning a student model by distilling the domain-robust knowledge from Progressive Translation.

8 Ethics Statement

The datasets used in the experiments are all well-known machine translation datasets and publicity available. Data preprocessing does not involve any external textual resources. Intermediate sequences generated in our data augmentation method are new symbolic combinations of the tokens in the target language. However, the final output of the model is the tgt sequence which is the same as the target sequence in the original training set. Therefore, we would not expect the model trained with our data augmentation method would produce more harmful biases. Finally, we declare that any biases or offensive contexts generated from the model do not reflect the views or values of the authors.

References


David Dale, Elena Voita, Loïc Barrault, and Marta R. Costa-jussà. 2022. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity even better.


Noam Wies, Yoav Levine, and Amnon Shashua. 2022. Sub-task decomposition enables learning in sequence to sequence tasks.


A Appendix

A.1 Discussion of Intermediate Sequences

`lex` and `ali` intermediate sequences may come from certain intermediate topological spaces between the transformation from the topological spaces of the source into the target languages. We empirically confirm that such intermediate sequences might look strange but are easier for the neural model to learn and predict, since they are structurally closer to the source. We use the standard Transformer model to learn to predict `lex`, `ali` and `tgt` (this is just the baseline) directly on IWSLT’14 dataset and report the results on both in-domain and out-of-domain test sets. Note that the gold-standard sequences of `lex` and `ali` on the out-of-domain test sets are produced on the corresponding out-of-domain training sets.

Table 5 shows that `lex` is easier to be predicted than `ali`, and `ali` is easier to be predicted than `tgt` by the NMT model, over both in-domain and out-of-domain test sets.

<table>
<thead>
<tr>
<th>Domain</th>
<th>lex</th>
<th>ali</th>
<th>tgt</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>94.0±0.20</td>
<td>61.1±0.12</td>
<td>32.1±0.38</td>
</tr>
<tr>
<td>OOD</td>
<td>72.6±0.60</td>
<td>47.9±0.48</td>
<td>13.9±0.19</td>
</tr>
</tbody>
</table>

Table 5: Average BLEU (↑) and standard deviation on in-domain and out-of-domain test sets on IWSLT’14 DE→EN when the target is `lex`, `ali` or `tgt` separately.
## A.2 Hyperparameters

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>IWSLT</th>
<th>OPUS/Allegra</th>
</tr>
</thead>
<tbody>
<tr>
<td>embedding layer size</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>hidden state size</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>tie encoder decoder embeddings</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>tie decoder embeddings</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>loss function</td>
<td>per-token-cross-entropy</td>
<td></td>
</tr>
<tr>
<td>label smoothing</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>optimizer</td>
<td>adam</td>
<td></td>
</tr>
<tr>
<td>learning schedule</td>
<td>transformer</td>
<td></td>
</tr>
<tr>
<td>warmup steps</td>
<td>4000</td>
<td>6000</td>
</tr>
<tr>
<td>gradient clipping threshold</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>maximum sequence length</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>token batch size</td>
<td>4096</td>
<td></td>
</tr>
<tr>
<td>length normalization alpha</td>
<td>0.6</td>
<td>1</td>
</tr>
<tr>
<td>encoder depth</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>decoder depth</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>feed forward num hidden</td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>number of attention heads</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>embedding dropout</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>residual dropout</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>relu dropout</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>attention weights dropout</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>beam size</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>validation frequency</td>
<td>4000 iterations</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Configurations of NMT systems over three datasets.
ACL 2023 Responsible NLP Checklist

A  For every submission:

☑ A1. Did you describe the limitations of your work? 7

☑ A2. Did you discuss any potential risks of your work? 8

☑ A3. Do the abstract and introduction summarize the paper’s main claims? 1

☒ A4. Have you used AI writing assistants when working on this paper? Left blank.

B  Did you use or create scientific artifacts?

4.1

☑ B1. Did you cite the creators of artifacts you used? 4.1

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts? It has been provided in the cited work.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)? Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it? Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.? 4.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be. 4.1

C  Did you run computational experiments?

4

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used? Our model is the standard Transformer model without any modification to the model’s architecture.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.
C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4.2

C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4.3

C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4.1

D  ✅ Did you use human annotators (e.g., crowdworkers) or research with human participants?
5.1

D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
 Instructions are rather simple which could be indicated in section 5.1.

D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants’ demographic (e.g., country of residence)?
 Left blank.

D3. Did you discuss whether and how consent was obtained from people whose data you’re using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
 Not applicable. Left blank.

D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
 Left blank.

D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
 Left blank.