Improving Chinese-Vietnamese Neural Machine Translation with Linguistic Differences

ZHIQIANG YU, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan Minzu University, Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, China
ZHENGTAO YU, YANTUAN XIAN, YUXIN HUANG, and JUNJUN GUO, Faculty of Information Engineering and Automation, Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, China

We present a simple, efficient data augmentation approach for boosting Chinese-Vietnamese neural machine translation performance by leveraging the linguistic difference between the two languages. We first define the formalized representation of modifier symmetry, which is one of the most representative linguistic differences between Chinese and Vietnamese. We then propose and test two data augmentation strategies for leveraging the linguistic difference, which can be integrated naturally with different translation models. Results indicate that both strategies can introduce linguistic rules to boost translation accuracy. Tests on Chinese-Vietnamese benchmarks show significant accuracy improvements. To facilitate studies in this domain, we also release an open-source toolkit[1] with flexible implementation for Chinese-Vietnamese linguistic difference tagging.

CCS Concepts: • **Computing methodologies** → **Machine translation**;

Additional Key Words and Phrases: Neural machine translation, Chinese-Vietnamese, linguistic difference, data augmentation

[1] https://github.com/yzqyt/ldnmt.

The work is supported by the national key research and development plan project (Grant No. 2019QY1800), National Natural Science Foundation of China (Grant Nos. 61732005, 61672271, 61761026, 61762056, and 61866020), Yunnan provincial major science and technology special plan projects (Grant No. 202002AD080001), Yunnan high-tech industry development project (Grant No. 201606), and Natural Science Foundation of Yunnan Province (Grant No. 2018FB104).
Authors' addresses: Z. Yu, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan Minzu University, Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, No. 727 South Jingming Rd, Chenggong District, Kunming, Yunnan, 650500, China; email: yzqyt@hotmail.com; Z. Yu (corresponding author), Y. Xian, Y. Huang, and J. Guo, Faculty of Information Engineering and Automation, Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, No. 727 South Jingming Rd, Chenggong District, Kunming, Yunnan, 650500, China; emails: ztyu@hotmail.com, xianyt@kust.edu.cn, {huangyuxin2004, guojjgb}@163.com.

2375-4699/2022/03-ART22 $15.00
https://doi.org/10.1145/3477536

ACM Reference format:
Zhiqiang Yu, Zhengtao Yu, Yantuan Xian, Yuxin Huang, and Junjun Guo. 2022. Improving Chinese-Vietnamese Neural Machine Translation with Linguistic Differences. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 21, 2, Article 22 (March 2022), 12 pages.
https://doi.org/10.1145/3477536

1 INTRODUCTION

Neural machine translation (NMT) has achieved high-quality performance in high-resource data conditions [1–3]. However, in low-resource conditions, the performance of NMT drops starkly, frequently underperforming traditional statistical machine translation (SMT) and requiring large amounts of training data to achieve competitive results [4–6]. To address the data sparsity problem in low-resource conditions, a number of data augmentation approaches have been proposed. Although these significant works enhanced the scale and quality of the available parallel corpus [7–10], linguistic rules are difficult to handle and integrate into NMT at runtime.

Chinese-Vietnamese is a typical low-resource language pair. Limited by the scale of a parallel corpus and the supply of Vietnamese language processing tools, dominant low-resource NMT approaches show suboptimal performances on Chinese-Vietnamese language pairs [11, 12]. However, Chinese and Vietnamese have considerable linguistic differences, the most significant one being the reverse order of modifiers. For example, for the Chinese phrase "*最*(the most) *美 丽 的*(beautiful) *女 孩*(girl)," its corresponding Vietnamese phrase is "*cô gái*(girl) *đẹp*(beautiful) *nhất*(the most)." The inverse modifier order is an important linguistic rule for Chinese-Vietnamese translation, but there is still a lack of enough approaches to apply it to NMT. Intuitively, for using this linguistic difference, the following preconditions need to be met: (1) formalized representation—the linguistic difference should be recognized easily and represented as modifier sequences and is easier to feed into dominant NMT frameworks; and (2) flexible integration approach—an efficient and flexible approach should be used to integrate the modifier sequences into NMT input. To tackle these problems, we recognize and represent the linguistic difference as a modifier sequence and propose a novel data augmentation approach to generate a new corpus based on it. Our main contributions are as follows:

- We investigate the characteristics of the inverse modifier order between Chinese and Vietnamese from the perspective of linguistic differences and discuss the feasibility of extracting the character into the linguistic rule sequence. For formalized representation, we release an open-source Chinese-Vietnamese linguistic difference tagging toolkit with flexible modular implementation.
- For utilizing formalized linguistic differences, we propose a novel data augmentation approach that allows linguistic differences to be integrated into the input sentences and generate a new corpus. We propose and test two simple but efficient strategies for the integration process.

The rest of the article is arranged as follows. We introduce the background of linguistic differences between Chinese and Vietnamese in Section 2. In Section 3, we present the details of our proposed approach in two phases: language difference recognition and data augmentation. We report the experimental process and results for our approach in Section 4. In Section 5, we conclude the article with an overview and potential future work.

2 LINGUISTIC DIFFERENCES

Chinese and Vietnamese are languages that belong to different language families. Unlike homologous language pairs, such as English-French and Thai-Lao, their pronunciation and writing are

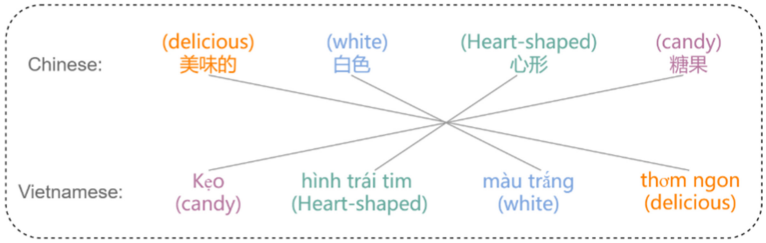

Fig. 1. Modifier symmetry in Chinese-Vietnamese. The words connected by solid lines are corresponding.

Table 1. Main Modifier Sequences of Modifier Symmetries

Modifier sequence	Modifier order	
	Chinese	Vietnamese
Noun	[*adverb*]-*adjective-noun* ([*d*]-*a*-*n*)	*noun-adjective*-[*adverb*] (*N*-*A*-[*R*])
Verb	*adverb-verb* (*d*-*v*)	*verb-adverb* (*V*-*R*)
Adjective	*adverb-adjective* (*d*-*a*)	*adjective-adverb* (*A*-*R*)

Note: [] indicates optional; contents in parentheses are the corresponding part of speech (POS) tag sequence.

completely different,[2] and spoken Chinese and Vietnamese are not mutually intelligible [13–15]. There are obvious linguistic differences in terms of language features. In the example in Figure 1, for the modification of the center word *candy*, the attributive that modifies the center word in Chinese is always prepositive, while the attributive in Vietnamese is usually postpositive. In Chinese, the basic descriptive attributives are (1) predicate phrase, (2) verb (phrase)/preposition phrase, (3) adjective phrases and other descriptive phrases, and (4) adjectives and descriptive nouns without "的." In the conventional condition, the order of descriptive attributive structures in Chinese is 1-2-3-4-center word, while for Vietnamese, the order is center word-4-3-2-1. The order of descriptive attributive structures in Vietnamese and Chinese is mirror-symmetric. This is a very important factor in translating Chinese to Vietnamese and vice versa.

In contrast to the postposition of the attributive in English, the postposition of the modifier is more obvious in Vietnamese [16]. In particular, the position of the modifier is symmetrical with that of Chinese, which forms a widely found linguistic difference in the Chinese-Vietnamese language pair. Based on specific linguistic analysis, we can describe a universal linguistic rule as follows: once there are modifiers in sequence for modifying, restricting or describing a center word in Chinese, the modifiers with the same semantics will almost certainly appear in Vietnamese. The difference is that modifiers are situated before the center word in Chinese, while in Vietnamese, it is inverse. We denote the linguistic difference rule as **modifier symmetry**. Table 1 illustrates the three kinds of main modifier sequences that represent modifier symmetry: noun, verb, and adjective. Take the noun modifier sequence as an example: the modifier order in Chinese is [*adverb*]-*adjective-noun*, while in Vietnamese, the order of modifiers is *noun-adjective*-[*adverb*].

3 OUR APPROACH

We present the details of our proposed approach in this section. Section 3.1 discusses language difference recognition; Section 3.2 presents the data augmentation approach in terms of realization.

[2]About 65% of Vietnamese words originate from Chinese and are often called Sino-Vietnamese words. However, with the transition of language, scripts in Vietnamese no longer use ones in Chinese.

Table 2. Main Part of Speech (POS) Tags in Chinese and Vietnamese

POS	ICTCLAS	VLSP
Noun	***n* (*n, ng, nr, nrfg, nrt, ns, nt, nz*)**	***N* (*N, Np, Nc, Nu, Ny, Nb*)**
Verb	***v*(*v, vd, vg, vi, vn, vq*)**	***V*(*V, Vb*)**
Adjective	***a***	***A***
Adverb	***d***	***R***
Pronoun	*r*	*P*
Conjunction	*c*	*C*(*C, Cc*)
Preposition	*p*	*E*
Numeral	*m*(*m, mg, mq*)	*M*

Note: We chose the most representative POS tags (noun, verb, adjective, and adverb) for modifier symmetry recognition.

ALGORITHM 1: Chinese Modifier Sequence Retrieval. [] means optional.

Input: Segmented Chinese sentence $x = (x_{1\ldots m})$, m is the sequence length
$tx \leftarrow$ POS(x), get tag sequence by POS tagging
$tx \leftarrow$ preprocess the tag sequence (e.g., convert tag $\in \{ng,\ nr,\ nrfg,\ nrt,\ ns,\ nt,\ nz, r\}$ to n,
convert (a, uj) tag to a, clean irrelevant tags, etc.)
for $i \leftarrow 1$ **to** m **do**
  if $tx_i = n$
    for $k \leftarrow 1$ **to** $i - 1$ **do**
      if $tx_{i-k} = a \vee (tx_{i-k} = d \wedge tx_{i-k+1} = a)$
        pass
      else if $tx_{i-k} \notin \{a, d\} \vee (tx_{i-k} = a \wedge tx_{i-k+1} = d)$
        clean tags $tx_{<i-k}$ and $tx_{>i}$, determine whether tx matches pattern [d]-[a]-a-n
return tx

3.1 Linguistic Difference Recognition

We summarized one of the most significant language differences between Chinese and Vietnamese as modifier symmetry. For integrating it into the NMT model, we need to describe it in a formalized representation. To this end, we describe modifier symmetry by POS tags. Specifically, standard POS tagging sets, ICTCLAS[3] and VLSP,[4] are chosen to represent Chinese and Vietnamese tagging, respectively. Table 2 lists the main POS tags in the two languages. To recognize modifier symmetry efficiently, we selectively use the most representative POS tags—noun, verb, adjective, and adverb—for modifier symmetry recognition.

Modifier symmetry recognition consists of two stages: (1) modifier sequence retrieval and (2) symmetry recognition. As the basis of symmetry recognition, modifier sequence retrieval first retrieves specific noun, verb, and adjective modifier sequences by linguistic rule matching. Take the Chinese noun modifier sequence retrieval shown in Algorithm 1 as an example. Given the input sentence $x = (x_{1\ldots m})$ with length m, POS tagging and preprocessing operations are conducted on it, generating the POS tag sequence tx. Subsequently, tx is fed into the noun-regularized matching block. When the POS tag of tx at time i is n, we judge whether n is preceded by ([d], [a], a). Any sequence that does not conform to the linguistic rules will not be recorded. Chinese verb and adjective modifier sequence retrievals are done in the same manner as in Algorithm 1. The implementation details, such as how to distinguish a noun modifier sequence comprising an adverb

[3] http://ictclas.nlpir.org/.
[4] http://vlsp.vietlp.org.

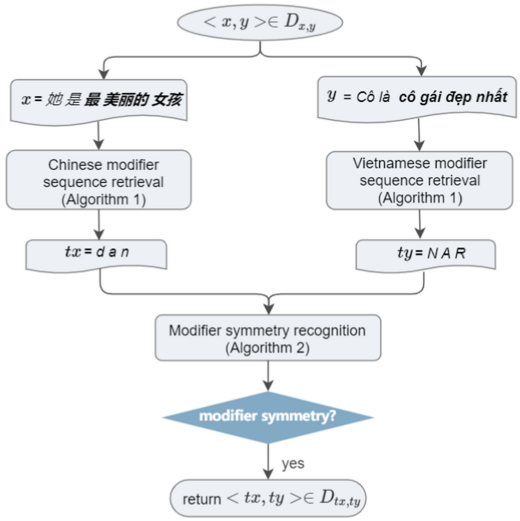

Fig. 2. A running example of executing Algorithm 1 and Algorithm 2.

ALGORITHM 2: Modifier Symmetry Recognition

Input: Chinese-Vietnamese training data corpus $D_{x,y}$
while $x, y \neq EOF$ **do**
  $x, y \leftarrow$ traverse $D_{x,y}$, get parallel sentence pairs
  $x,\ y \leftarrow$ segment x and y
  $tx \leftarrow$ get Chinese modifier sequence (e.g., (…d, a, n…)) by x, where …is the empty tag
  $ty \leftarrow$ get Vietnamese modifier sequence by y (e.g., (…N, A, R…))
  $ty_{zh} \leftarrow$ Chinese format copy of ty (e.g., (…N, A, R…) to (…d, a, n…))
  if $tx = ty_{zh}$
    record tx and ty
return $D_{tx,ty}$

(*d, a, n*) and adjective modifier sequence (*d, a*) through linguistic rules, can be found in the supplementary source codes. Vietnamese modifier retrieval is analogous to Chinese modifier retrieval. The slight differences are the specific linguistic rules and the representation of returned tags. We omit the redundant algorithm descriptions for the sake of simplicity.

The modifier symmetry recognition algorithm is presented in Algorithm 2. Given the input Chinese-Vietnamese corpus $D_{x,y}$, a simple iterator first traverses it and gets the parallel segmented Chinese sentence x and Vietnamese sentence y. Then, the modifier tag sequences tx and ty are generated by the respective retriever based on x and y. After a simple equality judgment, we record the qualified tx and ty into $D_{tx,ty}$ and return it. $D_{tx,ty}$ and original corpus $D_{x,y}$ are the input of the downstream data augmentation approach described in Section 3.2. Figure 2 illustrates a running example of executing Algorithm 1 and Algorithm 2 for the Chinese-Vietnamese input sentence pair <*她 是 最 美丽的 女孩* (*She is the most beautiful girl*), *Cô là cô gái đẹp nhất* (*She is girl beautiful the most*)> $\in D_{x,y}$. We ultimately get its corresponding modifier sequence pair $\langle (d,\ a,\ n), (N, A, R) \rangle \in D_{tx,ty}$.

3.2 Data Augmentation

We propose a data augmentation approach in which the NMT model learns how to use the original corpus $D_{x,y}$ and modifier symmetry representation $D_{tx,ty}$ obtained from Section 3.1.

Fig. 3. The two alternative strategies used to generate source sentences of training data. For unified tagging, *o, s, t* denote source words, source modifier block, and target modifier block, respectively. For specific tagging, *o* denotes source words; others are the original POS tags copied from the source and target sentences.

Inspired by efficient works [17, 18] that implement data augmentation by integrating additional information into traditional NMT input, we opt here for integrating the modifier symmetry as in-line annotations in the source sentence. Take the parallel sentence pair <*她 是 最 美丽的 女孩*, *Cô là cô gái đẹp nhất*> $\in D_{x,y}$. For example, we first integrate its corresponding modifier sequence pair $\langle (d, a, n), (N, A, R) \rangle \in D_{tx,ty}$ into it and generate the tagged sentence pair <*她 是 最* $_d$ *美丽的* $_a$ *女孩* $_n$, *Cô là cô gái*$_N$ *đẹp*$_A$ *nhất*$_R$> illustrated in Figure 3. Then, we add tags to signal the origin of the word and generate a new source sentence. There are two tagging strategies: (1) unified tagging, in which tags *o, s*, and *t* denote source words, source modifier block, and target modifier block, respectively; and (2) specific tagging, in which tag *o* denotes source words and others are the original POS tags copied from source and target sentences. As shown in Figure 3, the POS tags are added as an additional parallel stream to signal the origin of each word in the source sentence. We record the augmented corpus as $D_{x',y'}$, which serves as the input for NMT model training together with $D_{x,y}$.

As we do not change the original sequence-to-sequence NMT framework, the network can learn the utilization of language difference from the augmentation of the training data. However, the word alignment information that we use implicitly is from the original corpus. To investigate whether our approach based on implicit word alignment can achieve the desired performance, we apply strict word alignment verification by lexicographic matching to the recognized result and conduct contrast experiments in Section 4.

4 EXPERIMENTS

In this section, we present empirical studies for the proposed approach to the Chinese-Vietnamese NMT task. We also conduct multiple studies to thoroughly analyze the effect of the proposed approach, including fluency, copy behavior, and case study.

4.1 Data

Parallel data: We evaluate our approach to the Chinese-Vietnamese translation task by training models on the ALT dataset and our inhouse dataset.

- **ALT Chinese-Vietnamese.** The ALT dataset is a small-sized multilingual parallel dataset supplied by the Asian Language Treebank Project.[5] According to our need, we select the Chinese-Vietnamese subset and bin it by the auxiliary splitting tool ALT-Standard-Split.

[5] http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/.

The preprocessed dataset comprises 18,088, 1,000, and 1,018 sentence pairs as training, development, and test sets respectively.

- **Inhouse.** The inhouse Chinese-Vietnamese dataset is built artificially, which comprises 100K parallel sentence pairs. We follow the partition ratio of IWSLT15 and cut 1,500 and 1,000 sentence pairs from the inhouse dataset as development and test sets, respectively. The rest are used for training.

Lexicon data: To fulfill the data requirement of data augmentation baseline systems, we build the Chinese-Vietnamese bilingual parallel lexicon. The lexicon comprises 51K word pairs; 70% of the word pairs are collected manually and the rest are obtained from dict.land.[6] Note that our approach does not rely on the external lexicon, whereas for the sake of fair comparison, we use the lexicon for the strict word alignment verification (see Section 4.5).

4.2 Baseline

We compare our approach with the strong SMT and NMT basic model and the homogeneous previous works:

- Moses: The dominant phrase-based SMT system with the default configuration and a 4-gram language model. We train the model on the entire training data. In low-resource settings, SMT tends to achieve better performance than NMT [19].
- Transformer [20]: The dominant approach that obtained significant performance on machine translation and predicts target sentence from left to right, relying on self-attention. We run Transformer as well as Moses on the default corpus without data augmentation.
- Code-switching [17]: A data augmentation approach that uses *replace* strategy for term lexicon introducing. For each source-target word, randomly sampling $k1$ matching sentences to replace a source-side word with its target-side word. For each combination of two source-target word pairs, the sampling hyper-parameter is set to $k2$ and both source-side matching words are replaced with their target translations.
- Term-constraint [18]: A data augmentation approach that utilizes an external lexicon on the generic NMT architecture. The approach provides two strategies: *append* and *replace* for data augmentation. For *append*, the target words in the lexicon are appended to the corresponding source words. For *replace*, the words in source sentences are replaced by the corresponding words in the lexicon. We compare our approach to the *append* strategy, which performs better according to the conclusion of the work.

4.3 Preprocessing and Training Details

We tokenize the datasets using our inhouse toolkit, which is high performance, and the source codes are included in the supplementary documents. For POS tagging, we use the POS module in the inhouse toolkit that is implemented based on jieba tools[7] and VnMarMoT.[8] To avoid confusion between English letters and POS tags, we do not use the BPE approach on the experimental parallel corpus. In addition, to prevent the Code-switching and Term-constraint model from degenerating into greedy replacement, we follow the instructions in [17, 18] and annotate only the word pairs that present in both source and reference.

We follow the guidelines of Sennrich and Zhang [19] on optimizing low-resource NMT and adopt prudent parameter settings. The vanilla Transformer model for running baseline and our

[6] https://www.dict.land/.
[7] https://github.com/fxsjy/jieba.
[8] https://github.com/datquocnguyen/VnMarMoT.

Table 3. Recognition Results of ALT and Inhouse Datasets

	Noun	Verb	Adjective	After sv
ALT	4.4%	5.6%	4.1%	94%
Inhouse	6.2%	6.9%	4.7%	93%

Note: "sv" denotes strict word alignment verification.

approach comprise 2-layer encoders and decoders, a shared source, and target embeddings. Moreover, the batch size is set to 128 and the dimensions of word embeddings, hidden states, and the filter sizes are set to 256, 256, and 512, respectively. We use the dropout of 0.1 and set label smoothing to 0.1 for regularization. For model optimization, we use the Adam [21] optimizer and set the learning rate to 10^{-3}. We stop training early when there is no improvement of the BLEU for 10 consecutive checkpoints. During inference, we use beam search with the beam size of 4 and length penalty of 1 for all datasets. The models are trained on one P100 GPU and are evaluated every 1,000 steps.

4.4 Evaluation

We choose the case-insensitive 4-gram BLEU score as the main evaluation metrics [22] and adopt the script multi-bleu.perl in the Mose toolkit.[9] Significance tests are conducted based on the best BLEU results by using bootstrap resampling [23]. For all datasets, we use the checkpoint, which has the best performance on the validation set. Checkpoint averaging is not adopted except for specific notification.

4.5 Experimental Results

Recognition rates. Table 3 shows the recognition results. The "noun," "verb," and "adjective" columns denote the recognition rates of the three kinds of modifier sequences, respectively, computed as the percentage of times the modifier sequence was recognized out of the original sentence pairs. "After sv" columns denote the rates of modifier sequences that meet strict word alignment verification, computed as the percentage of times the qualified modifier sequences out of the total modifier sequences.

The first observation we make is that the overall retrieval rate on the inhouse dataset is higher than the ALT dataset. A possible reason is that the inhouse corpus comprises a certain number of parallel sentences that cover the literary domain. Compared with ALT datasets, which mainly contain journalistic domains, sentences that belong to the literary domain are more likely to contain descriptive pieces. Moreover, we use the lexicon for the strict word alignment verification of data augmentation described in Section 3.2. Each modifier symmetry pair should not only meet the word alignment constraints but also pass the semantic checks to ensure that they are matched in the lexicon. Note that when checking for matches of the words inside a modifier symmetry pair, synonyms are also considered as successful matching. We also observe that at least 93% of modifier sequences passed strict verification. The results show that our approach can accurately recognize linguistic differences even without lexicon verification.

Translation Quality. Table 4 shows the experimental results evaluated by BLUE score. For the ALT dataset, our approach scores between 10.57 and 11.08 BLEU points higher than the best baseline system on the zh→vi translation task. There is a similar increasing trend on the vi→zh task. For the inhouse dataset, our approach outperforms the best baseline system on the zh→vi task and vice versa. In addition, we note that in our approach, using specific tagging (st) performs better than using unified tagging (ut).

[9] http://www.statmt.org/moses.

Table 4. Translation Quality Evaluation of ALT and Inhouse Datasets

Models	ALT		Inhouse	
	zh→vi	vi→zh	zh→vi	vi→zh
Moses	8.84	8.72	17.16	17.94
Transformer	8.51	8.44	17.88	18.36
Code-switching	9.23	9.09	18.90	18.77
Term-constraint	9.50	9.27	18.85	18.64
Our approach(ut)	10.57	9.81	19.73	19.10
Our approach(st)	10.76	10.08	19.90	19.17
Our approach(ut+sv)	10.84	10.25	19.95	19.21
Our approach(st+sv)	**11.08**	**10.43**	**20.19**	**19.42**

Note: "ut," "st," and "sv" denote unified tagging, specific tagging, and strict word alignment verification respectively.

Table 5. Fluency Evaluation on ALT Dataset

Evaluation	Models				
	Moses	Transformer	Code-switching	Term-constraint	Our approach (st)
RIBES	70.44	70.72	71.63	71.85	73.34
Fluency	3.38	3.41	3.49	3.52	3.81

As the baseline models, Code-switching and Term-constraint are lexicon based; we run them on the artificially built lexicon. For a fair comparison, we also use the lexicon to make a strict word alignment verification (sv) in our approach. As shown in Table 3, due to the strict verification on the modifier symmetry annotation, suboptimal matches (about 7%) are not utilized for new corpus building (see rows 8 and 9). We observe that using strict word alignment verification brings performance improvement compared with not using it. The results are also consistent with the conclusion in [19] that noise avoidance is very important in low-resource NMT. Although the strict word alignment verification causes about 7% loss of the modifier sequence, it still results in a performance boost. A possible reason is that our approach retains more than 93% of the modifier sequences, which are accurate and can be used for providing guidance information in the model training. The results demonstrate that our approach is robust and works well without external resources, such as lexicons.

Fluency. To investigate the effect of our approach on translation fluency, we evaluate our model by automatic evaluation (RIBES [24]) and manual evaluation (Fluency [25]), respectively. According to Table 5, for RIBES evaluation, our approach outperforms Transformer by 2.62 points, Code-switching by 1.71 points, and the Term-constraint model by 1.49 points. For manual evaluation, we randomly sample 300 source language sentences from the test set and ask three evaluators with knowledge of Chinese and Vietnamese linguistics to evaluate the translations without being told which system the translations are from. As shown in Table 5, our approach outperforms the best baseline system Term-constraint by 0.29 Fluency points. We observe that compared with the BLEU improvement reported in Table 4, our method shows a more significant improvement in fluency evaluation, which demonstrates that our approach can not only guide the generation of the target words but also makes the modifier sequence in the translation more accurate by rule-level constraints.

Copy behavior. The data augmentation baseline models based on lexicon integration encourage the integrated word in the source to be copied in the generated translation, while our

Table 6. Copy Behavior Evaluation on ALT and Inhouse Datasets

Dataset	Total			Compliance with Modifier Symmetry		
	baseline	ut	st	baseline	ut	st
ALT	61.3%	59.9%	54.2%	—	84%	88%
Inhouse	64.9%	63.7%	59.4%	—	81%	87%

Note: "ut" and "st" denote unified tagging and specific tagging, respectively. "Compliance with modifier symmetry" denotes the rates of word sequences that comply with modifier symmetry out of all output words.

Table 7. Example of Chinese-Vietnamese Translation Task

Input:	这是 (this is) 一场 (a) 非常 (very) 艰苦的 (tough) 战争 (war)
Reference:	Đây là (this is) một (a) trận chiến (war) gian khổ (tough) rất (very)
Baseline (Transformer):	Đây là (this is) một (a) cuộc chiến (fight) rất (very) khókhan (hard)
Our approach (st):	Đây là (this is) một (a) trận chiến (war) khókhan (hard) rất (very)
Input:	一束 (a) 新鲜的 (fresh) 红 (red) 花 (flower) 在窗口中 (in the window)
Reference:	Một (a) bông hoa (flower) màu đỏ (red) tươi (fresh) trong cửa sổ (in the window)
Baseline (Transformer):	Một (a) bông hoa (flower) tươi (fresh) màu đỏ (red) trong cửa sổ (in the window)
Our approach (st):	Một (a) bông hoa (flower) màu đỏ (red) tươi (fresh) trong cửa sổ (in the window)

Note: We use the underline fonts to indicate the different translations of the modifier sequence in baseline and our approach.

approach encourages the copy behavior explicitly by integrating modifier sequences. We conduct relevant analysis on the zh→vi translation task to investigate the difference in copy behavior. Specifically, we record the words added in the input sentence and compute how often they appear in the translated sentence. As seen in the columns under "Total" in Table 6, the number of words output by our approach (ut or st) is lower than those output by the baseline. However, the rates of word sequences in the output that comply with modifier symmetry after translation are over 81%. Considering that the output of translation in the NMT procedure is probability based, the rates are satisfactory. Considering the result of BLEU and fluency evaluation, we observed that the model can achieve better performance and fluency with a lower copy rate compared with baseline. A possible reason is that the integration of modifier sequences enforces the model to focus on the translation of modifier pieces, which is well suited for Chinese-Vietnamese translation task.

Case Study. Apart from the quantitative analysis, we provide a translation example of our proposed approach. In the first example in Table 7, the Chinese word "*战争*" (*war*) is translated more accurately in our approach compared with the baseline model Transformer. Moreover, the Transformer baseline model translates the Chinese modifier sequence "*非常* (*very*) *艰苦的* (*tough*)" to "*rất* (*very*) *khó khan* (*hard*)," while our model outputs "*khó khan* (*hard*) *rất* (*very*)." Compared with the former, the latter sentence is a more human-acceptable translation that is consistent with the linguistic rules of Vietnamese. The main possible reason for this phenomenon is that our approach provided clear guidance for the generation of modifier sequences. As the length of the sample sentence is short, there are only slight differences—*cuộc chiến* (*fight*) and *trận chiến* (*war*)—between

the two translations. The improvement of BLEU will not be very significant, whereas the translation generated by the latter is undoubtedly more consistent with the real expression. Similar translation behaviors can also be observed in the second example.

5 CONCLUSIONS

This article explores a data augmentation approach for boosting low-resource NMT performance by using linguistic differences. We choose modifier symmetry, one of the most significant features of language differences between Chinese and Vietnamese, for the representation. Based on the recognition algorithm that we proposed, we recognize the modifier symmetry efficiently and integrate it into the data augmentation procedure. In addition, two different tagging strategies are proposed to integrate modifier sequences into the original corpus. We perform data augmentation before training with negligible calculation overhead and release an open-source toolkit with an efficient and modularized implementation. Experimental results show that our approach can achieve significant improvement in the Chinese-Vietnamese translation task compared with strong transformer baseline systems and previous homogeneous works.

In our future work, we plan to optimize the linguistic rules used in the recognition algorithm in order to represent language differences more accurately. Formalized expression of other representative linguistic differences between the two languages is also our concern. In addition, we would like to investigate better techniques to integrate information of language differences, such as NMT structural modification techniques.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems -Volume 2 (NIPS'14)*. MIT Press, Cambridge, MA, USA, 3104–3112.

[2] D. Bahdanau, K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. arXiv.2014. 1409.0473.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems* 30, 5998–6008.

[4] P. Koehn and R. Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the 1st Workshop on Neural Machine Translation*. Vancouver, Canada, 2017, 28–39.

[5] Z. Tan, S. Wang, Z. Yang, G. Chen, X. Huang, M. Sun, and Y. Liu. 2021. Neural machine translation: A review of methods, resources, and tools. arXiv:2012.15515.

[6] Z. Yu, Z. Yu, J. Guo, Y. Huang, and Y. Wen. 2019. Efficient low-resource neural machine translation with reread and feedback mechanism. *ACM Transactions on Asian and Low-Resource Language Information Processing* 19, 3, Article 34 (2019), 13 pages. https://doi.org/10.1145/3365244

[7] M. Xia, X. Kong, A. Anastasopoulos, and G. Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy, July 28–August 2, 2019, 5786–5796.

[8] F. Burlot and F. Yvon. 2019. Using monolingual data in neural machine translation: A systematic study. arxiv:cs.CL/1903.11437.

[9] K. Song, Y. Zhang, H. Yu, W. Luo, K. Wang, and M. Zhang. 2019. Code-Switching for Enhancing NMT with Pre-Specified Translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota*. Association for Computational Linguistics, 449–459.

[10] D. Georgiana, M. Prashant, F. Marcello, and A. Yaser. 2019. Training neural machine translation to apply terminology constraints. arxiv:cs.CL/1906.01105.

[11] S. Gao, J. Huang, M. Xue, Z. Yu, Z. Wang, and Y. Zhang. 2019. Syntax-based chinese-vietnamese tree-to-tree statistical machine translation with bilingual features. *ACM Transactions on Asian and Low-Resource Language Information Processing* 18, 4, 36 (2019), 20. DOI: https://doi.org/10.1145/3314938

[12] W. Che, Z. Yu, Z. Yu, Y. Wen, and J. Guo. 2020. Towards integrated classification lexicon for handling unknown words in Chinese-Vietnamese neural machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing* 19, 3 (2020), 42 (2020), 17 pages. DOI: https://doi.org/10.1145/3373267

[13] A. T. Huu, H. Huang, and S. Shi. 2018. Integrating pronunciation into Chinese-Vietnamese statistical machine translation. *Tsinghua Science and Technology* 23, 6 (2018), 715–723.

[14] A. T. Huu, H. Huang, and S. Shi. 2019. Preordering for Chinese-Vietnam statistical machine translation. *IEICE Transactions on Information and Systems* E102-D, 2, 375–382.

[15] A. T. Huu, P. Tran, D. Dinh, V. V. Vu, and T. Le. 2018. Dependency-based pre-ordering of preposition phrases in Chinese-Vietnamese machine translation. *ICIC Express Letters, Part B: Applications* 9 (2018), 265–272.

[16] J. He, Z. Yu, C. Lv, H. Lai, S. Gao, and Y. Zhang. 2017. Language post positioned characteristic based Chinese-Vietnamese statistical machine translation method. In *Proceedings of the International Conference on Asian Language Processing (IALP'17)*, Singapore. IEEE, 2017.

[17] K. Song, Y. Zhang, and H. Yu. 2019. Code-switching for enhancing NMT with pre-specified translation[C]. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'19)*. Minneapolis, Minnesota, 2019.

[18] G. Dinu, P. Mathur, M. Federico, and Y. Al-Onaizan. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics, 3063–3068.

[19] R. Sennrich and B. Zhang. 2019. Revisiting low-resource neural machine translation: A case study[C]. In *57th Annual Meeting of the Association for Computational Linguistics (ACL'19)*. Florence, Italy. Association for Computational Linguistics, 211–221.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). 30. 5998–6008.

[21] D. P. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. arXiv:1412.6980V5.

[22] K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA*. Association for Computational Linguistics, 311–318.

[23] K. Philipp. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

[24] H. Isozaki, T. Hirao, and K. Duh. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing Cambridge, MA*. Association for Computational Linguistics, 944–952.

[25] M. Snover, N. Madnani, B. J. Dorr, and R. Schwartz. 2009. Fluency, adequacy, or HTER?: Exploring different human judgments with a tunable MT metric. In *Proceedings of the 4th Workshop on Statistical Machine Translation (StatMT'09)*. Association for Computational Linguistics, Stroudsburg, PA, 259–268. http://dl.acm.org/citation.cfm?id=1626431.1626480.

Received April 2021; revised July 2021; accepted July 2021