doi: 10.20532/cit.2021.1005316

Improving Machine Translation Quality with Denoising Autoencoder and Pre-Ordering

Tran Hong-Viet[1], Nguyen Van-Vinh[2] and Nguyen Hoang-Quan[2]

[1]University of Economic and Technical Industries, Hanoi, Vietnam
[2]University of Engineering Technology, Vietnamese National University, Hanoi, Vietnam

The problems in machine translation are related to the characteristics of a family of languages, especially syntactic divergences between languages. In the translation task, having both source and target languages in the same language family is a luxury that cannot be relied upon. The trained models for the task must overcome such differences either through manual augmentations or automatically inferred capacity built into the model design. In this work, we investigated the impact of multiple methods of differing word orders during translation and further experimented in assimilating the source languages syntax to the target word order using pre-ordering. We focused on the field of extremely low-resource scenarios. We also conducted experiments on practical data augmentation techniques that support the reordering capacity of the models through varying the target objectives, adding the secondary goal of removing noises or reordering broken input sequences. In particular, we propose methods to improve translation quality with the denoising autoencoder in Neural Machine Translation (NMT) and pre-ordering method in Phrase-based Statistical Machine Translation (PBSMT). The experiments with a number of English-Vietnamese pairs show the improvement in BLEU scores as compared to both the NMT and SMT systems.

ACM CCS (2012) Classification: Computing methodologies → Artificial intelligence → Natural language processing → Machine translation

Keywords: machine translation, phrase-based statistical machine translation, neural machine translation, pre-ordering, denoising autoencoder

1. Introduction

Two decades ago, Phrase-based Statistical Machine Translation (PBSMT) and then Neural Machine Translation (NMT) were the state-of-the-art methods for machine translation. Pre-ordering the source language sentences to match the target language word order was found useful in addressing word-order divergence for PBSMT [2, 3, 4, 5]. For NMT, Ponti *et al.* [6] and Yuki *et al.* [7] have explored pre-ordering methods and demonstrated that by reducing the syntactic divergence between the source and the target languages, consistent improvements in NMT performance can be achieved.

Besides these methods that attempted to reuse a well-investigated aspect of PBSMT, namely, pre-ordering, neural networks also have the option to improve their inherent reordering capacity for the task. However, as the interpretability of deep neural networks is quite low currently, our focus had to turn to augment training data, specifically with well-known techniques such as back-translation [28]. Our studies, however, showed that a comparable improvement can also be accomplished by adding the secondary objective of denoising sentences, which learns to remove heuristically added noises. This finding is consistent with other results that showed the denoising autoencoder is well suited to a variety of natural language processing (NLP) tasks concerning sentences [25, 33] and to improve poorly performing translation models [34].

In this paper, we focus on methods to improve the quality of machine translation with either denoising autoencoders or pre-ordering. Our main contributions are, as follows.

1. We investigated the language phenomena in English-Vietnamese machine translation for reordering problems, especially concerning these two languages.
2. We presented a supporting system for reordering and several variants of the autoencoder and corresponding data to aid the process of reordering target sentences, relying on the inherent structure of the target language learned during the denoising process.
3. We compared the results with the state-of-the-art data augmentation process and showed that the artificial noises created by the above denoising process are a suitable option to support the reordering capacity of translation models.

For reordering in PBSMT, Figure 1 shows one of the pre-ordering rules with an example sentence illustrating the effect of pre-ordering. Moreover, the context of words in the parallel source and assisting language sentences are similar, leading to consistent contextual representations across the source languages.

We also implemented a supporting system for reordering by utilizing denoising autoencoders to aid the process of reordering target sentences in NMT. The goal of this system was to apply reordering noises to NMT systems as an additional objective, as well as a supporting benchmark, to verify our claim that denoising is a useful secondary objective to our main problem.

This paper consists of six sections. Section 1 introduces the reordering problem, Section 2 provides an overview of the related work, Section 3 describes our methodology to investigate the language phenomena in English-Vietnamese translation. Section 4 presents our experimental results, Section 5 discusses the implication of the aforementioned results. Conclusions are given in Section 6.

2. Related Work

The word order problems are especially common for languages that have major differences, such as SOV (Subject-Object-Verb) vs. SVO (Subject-Verb-Object) languages, and cause insidious, but entirely avoidable errors for machine translation of the language pairs where the word order is almost right, but not quite so. This could lead to the neural network's attention mechanism in general and the decoder layers, in particular, generating different translations from the same sentence in the source and target language. This is an undesirable phenomenon because we want to transfer the knowledge from the parent model (assisting source/target) to the child model (source/target).

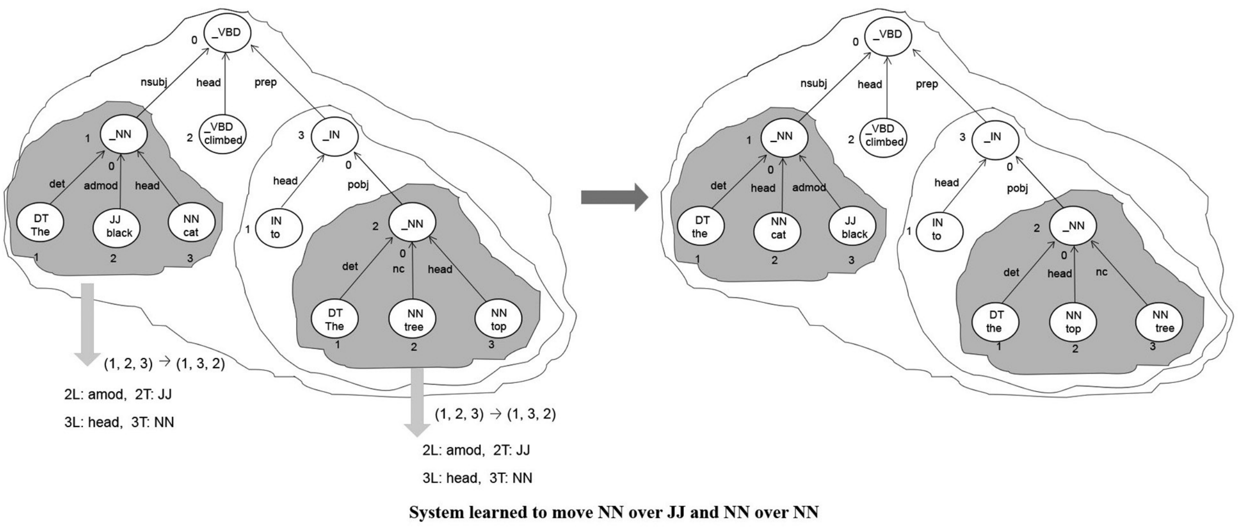


Figure 1. An example of reordering based on the dependency tree.

Many solutions to the reordering problem have been proposed, *e.g.*, syntax-based models [13], reordering [14], and tree-to-string methods [15]. Regarding syntax reordering methods, [13] presented a significant improvement through the incorporation of the existing strength of phrase and syntax into statistical machine translation (SMT). While Collins *et al.* [2] employed a parser tree, which is powerful in capturing sentence structures, other approaches applied reordering at the word level, which is more beneficial to richly morphological languages since it can help reduce data sparseness. Studies [16, 17, 18] considered the balance between translating quality and decoding time to employ the reordering method as a pre-processing step.

Besides that, Nguyen and Chiang [8] used Byte Pair Encoding (BPE) as basic input representation units. Research done by Lee *et al.* [9] applied character-level NMT system and Gu *et al.* [10] explored bilingual embedding. These studies have tried to address the lexical divergence between the source and the target languages. However, the effects of word order divergence and its mitigation have not been explored in detail.

In [25], a method was presented to learn distributed representations of sentences from unlabeled data; Xie *et al.* [33] used noising and denoising of natural language in a form of diverse back-translation for grammar correction; Yunsu *et al.* [34] presented effective cross-lingual transfer of NMT models without shared vocabularies. These studies used denoising autoencoders for a variety of NLP tasks concerning sentences to improve the poorly performing translation model.

3. Our Method

In this section, we will provide an analysis of the phenomena occurring in translation between English and Vietnamese, in which word segmentation, morphology, and word order are to be investigated. In particular, we focus on parts of the grammar of the language pair, specifically concerning the reordering issue, during translation from English to Vietnamese. These analyses would be used to further complement our experimental focus and explain how our methods achieve their results.

We implemented multiple systems for reordering, ranging from syntax-based and statistically based rule sets that would be applied to existing SMT (and by extension PBSMT) models; as well as two different ways of augmenting data aimed to mimic the noisy input received by NMT models due to syntactic differences between languages. From these options, we analyze the results achieved and identify the benefit of our implementations.

3.1. Pre-Ordering Methodology for English-Vietnamese Machine Translation

In the English-Vietnamese parallel corpus, the linguistics phenomena always have much more movement in word orders when compared to the same family of language pairs, which is not at all surprising. Usually, NMT systems require a sizable amount of parallel data to exhibit good results in this aspect, which means that before the training data reaches a needed threshold, differently structured sentences (*i.e.*, questions) being translated from the English source side into the Vietnamese target language would end up with artifacts that hamper translation quality. This is not the case for PBSMT systems trained on a similarly sized corpus, as it could solve the reordering problem better than NMT based on applying modifiable statistical rules during reordering. The power of PBSMT is in modeling short reordering and local context. However, as long-distance reordering is still a hard problem in phrase-based SMT, we propose a pre-ordering method to improve the quality of English-Vietnamese PBSMT machine translation, as detailed below.

For the input sentences in the source-side language, we apply the pre-ordering method in PBSMT for English-Vietnamese machine translation. The pre-ordering is a procedure to reorder a source sentence into the order of a target sentence by parsing the source sentence and then applying rules in both training and test data. This model has proven beneficial to achieve better translation performances in our method. Figure 2 shows an example of pre-ordering for English-Vietnamese machine translation.

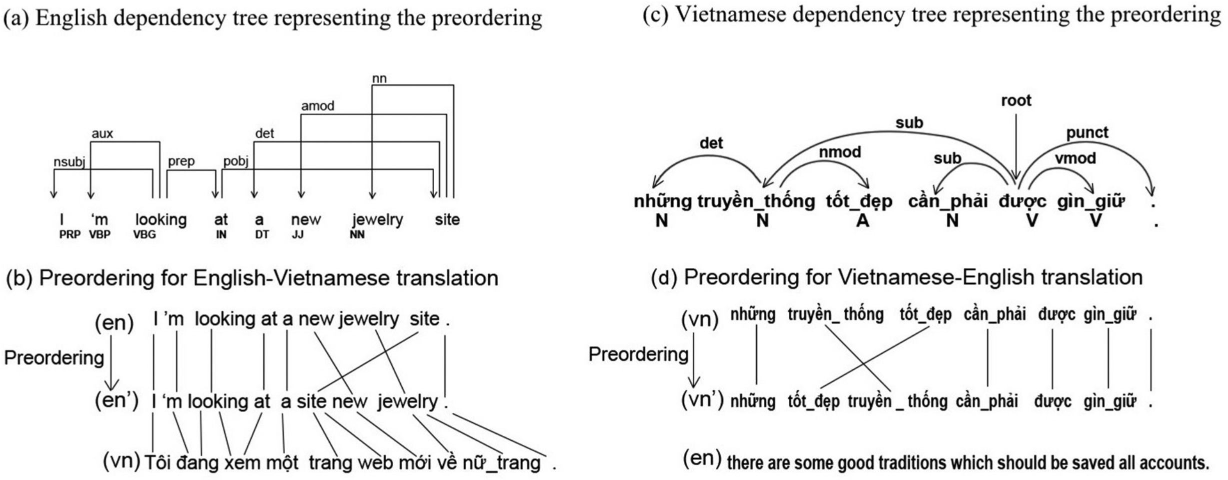

(a) English dependency tree representing the preordering
nn
amod
aux
det
nsubj
prep
pobj
I 'm looking at a new jewelry site
PRP VBP VBG IN DT JJ NN
(b) Preordering for English-Vietnamese translation
(en) I 'm looking at a new jewelry site .
Preordering
(en') I 'm looking at a site new jewelry .
(vn) Tôi đang xem một trang web mới về nữ_trang .
(c) Vietnamese dependency tree representing the preordering
root
sub
punct
det
nmod
sub
vmod
những truyền_thống tốt_đẹp cần_phải được gìn_giữ .
N N A N V V .
(d) Preordering for Vietnamese-English translation
(vn) những truyền_ thống tốt_đẹp cần_phải được gìn_giữ .
Preordering
(vn') những tốt_đẹp truyền _ thống cần_phải được gìn_giữ .
(en) there are some good traditions which should be saved all accounts.

Figure 2. Example of pre-ordering for English-Vietnamese machine translation.

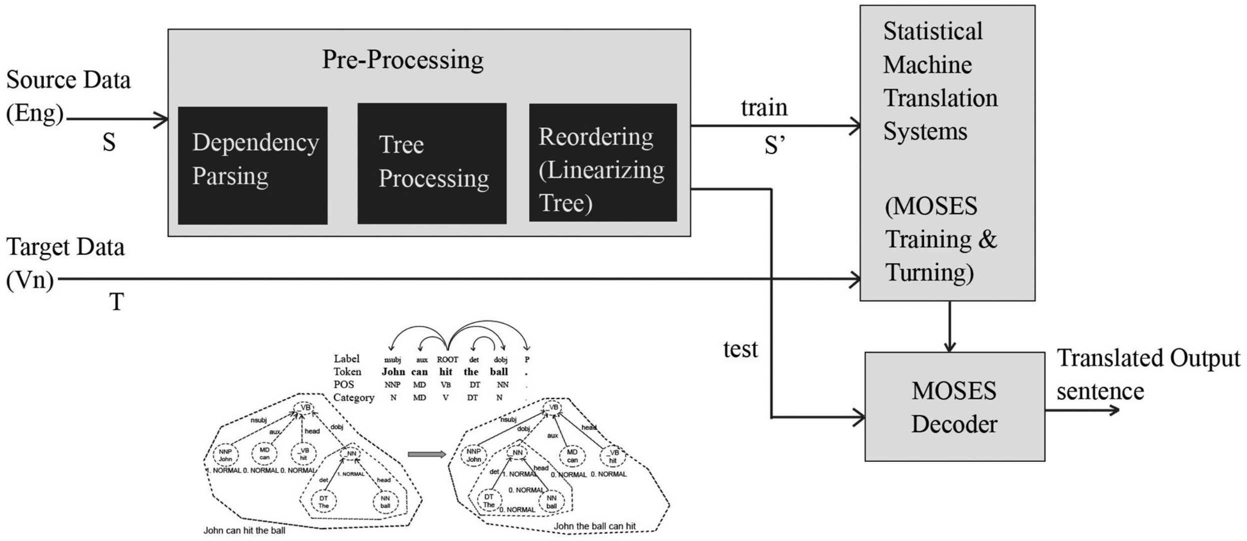
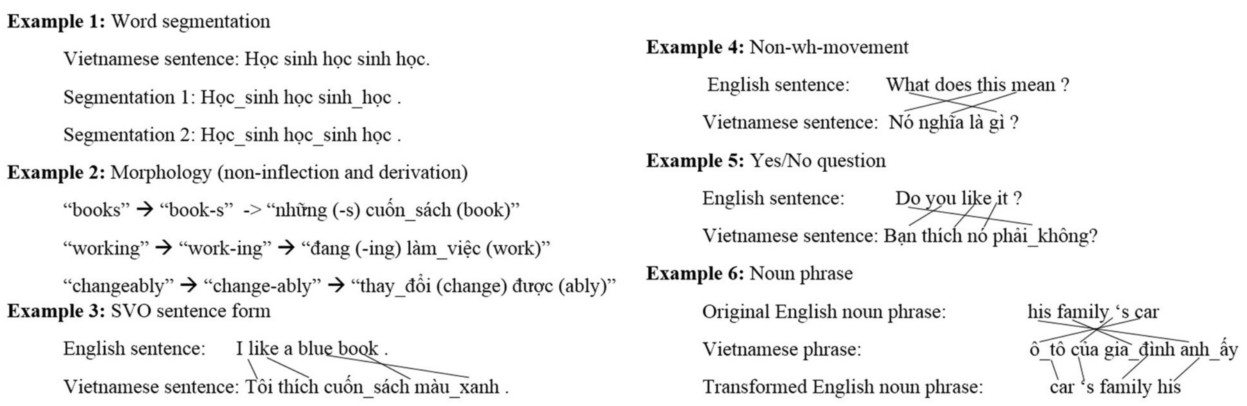


Figure 3. Pre-ordering method for PBSMT to improve reordering problem in PBSMT.



Example 1: Word segmentation

Vietnamese sentence: Học sinh học sinh học.

Segmentation 1: Học_sinh học sinh_học .

Segmentation 2: Học_sinh học_sinh học .

Example 2: Morphology (non-inflection and derivation)

"books" → "book-s" -> "những (-s) cuốn_sách (book)"

"working" → "work-ing" → "đang (-ing) làm_việc (work)"

"changeably" → "change-ably" → "thay_đổi (change) được (ably)"

Example 3: SVO sentence form

English sentence: I like a blue book .

Vietnamese sentence: Tôi thích cuốn_sách màu_xanh .

Example 4: Non-wh-movement

English sentence: What does this mean ?

Vietnamese sentence: Nó nghĩa là gì ?

Example 5: Yes/No question

English sentence: Do you like it ?

Vietnamese sentence: Bạn thích nó phải_không?

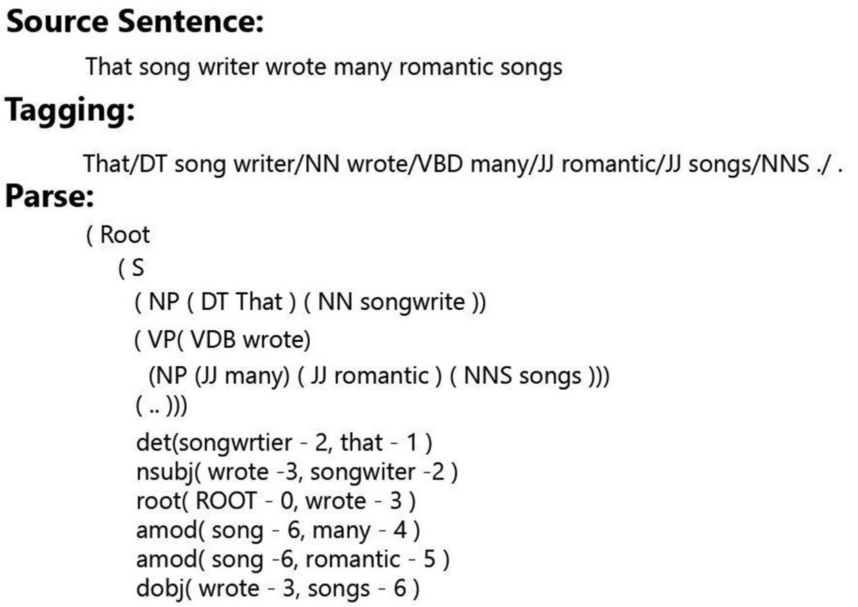
Example 6: Noun phrase

Original English noun phrase: his family 's car

Vietnamese phrase: ô_tô của gia_đình anh_ấy

Transformed English noun phrase: car 's family his

Figure 4. An example describes some phenomena specific to the Vietnamese language.



Source Sentence:

That song writer wrote many romantic songs

Tagging:

That/DT song writer/NN wrote/VBD many/JJ romantic/JJ songs/NNS ./ .

Parse:

```
( Root
   ( S
     ( NP ( DT That ) ( NN songwrite ))
     ( VP( VDB wrote)
       (NP (JJ many) ( JJ romantic ) ( NNS songs )))
     ( .. )))
     det(songwrtier - 2, that - 1 )
     nsubj( wrote -3, songwiter -2 )
     root( ROOT - 0, wrote - 3 )
     amod( song - 6, many - 4 )
     amod( song -6, romantic - 5 )
     dobj( wrote - 3, songs - 6 )
```

Figure 5. Example about Dependency Parser of an English sentence using Stanford Parser.

In this work, we used dependency trees for the pre-ordering method in PBSMT for English-Vietnamese machine translation. The pre-ordering process performs as shown in Figure 3.

Before applying the pre-ordering approach in PBSMT, we analyzed the phenomena between English and Vietnamese, in which word segmentation, morphology, and word order would be investigated. The phenomena-based linguistic analysis could help with providing improvements in translation performance by pointing out specific problems that still plague the translation process. As English and Vietnamese share the SVO verb order, we focus on the representative relations which consist of the noun phrase, adjectival phrase, adverbial phrase, and preposition and establish a set of reordering rules for English-Vietnamese language pair, for example, we often must move its pre-modifiers to follow the headword as both adjectives and adverbs belong to this reordering category. The differences are illustrated in Figure 4.

3.1.1. Pre-Ordering for PBSMT Using Reordering Rules Extracted from the Parallel Corpus

To create our set of reordering rules, we used dependency trees and the distinction of word order between English and Vietnamese. Based on POS-tags (Part-Of-Speech tags) and input sentence parsing results, the dependencies of POS-tags and head-modifier are shown as an example in Figure 5.

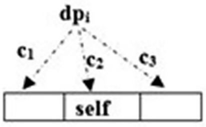
From the result of the relationship survey between the POS-tags type and dependency label's order in reordering word order, we use Algorithm 1 to generate rules based on survey corpus. These reordering rules are to be applied in the pre-ordering process before using PBSMT's translation module for the English-Vietnamese language.

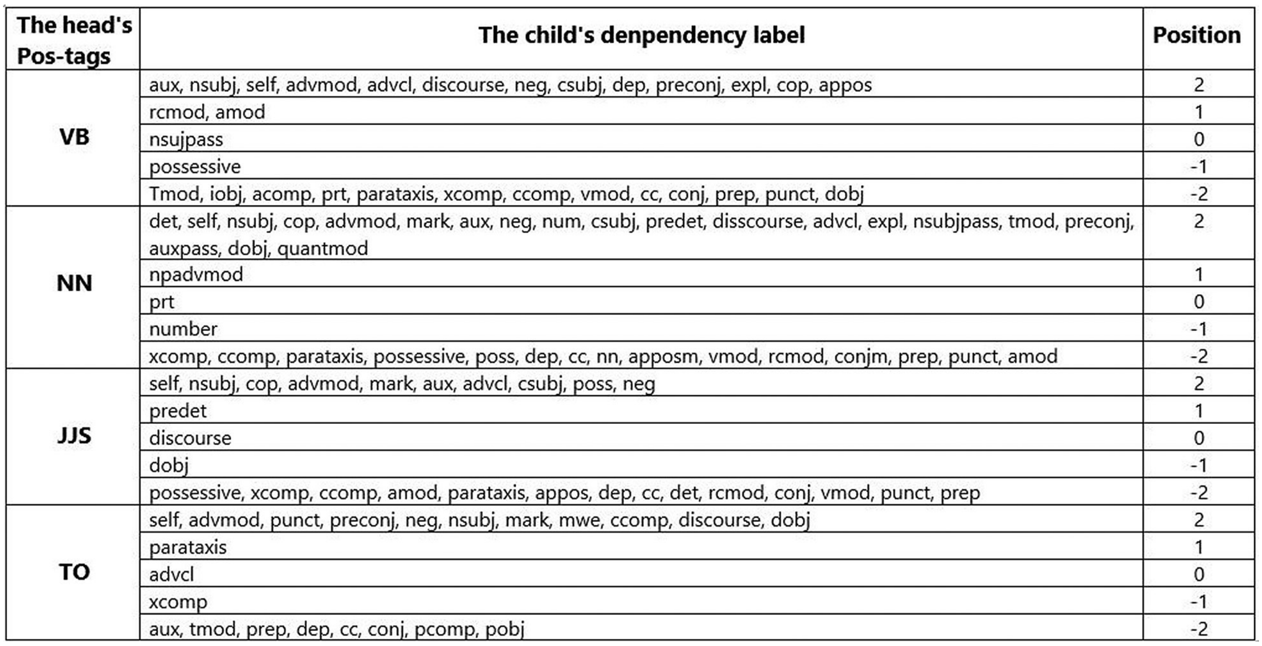
Our survey of the phenomena between English and Vietnamese is an important reason why we proposed our combined approach, which preserves the adequacy of PBSMT and the fluency of NMT to improve the quality of the translation system. Vietnamese is a non-inflectional language, while most English-inflected word forms can be translated into a Vietnamese phrase. The word form is analyzed morphologically to a lemma and an inflectional suffix. The lemma is translated into a Vietnamese word which is the head of the phrase, and the suffix into a Vietnamese function word which precedes/follows and modifies the headword. English derivative words often correspond to Vietnamese compound words. Vietnamese has a different word order from English. Figure 6 shows the result of our survey of the POS-tags type and dependency label's order in reordering word order. Our position values 2, 1, 0, −1, −2 correspond to the positions of the nodes from left to right after reordering, with the nodes to which larger positions were assigned always standing before nodes with smaller positions. Nodes with the same positions retain their ordering relative to each other. While technically we can assign more position values to the rules, we found that due to the syntactic similarity of the two languages these five values are sufficient for our reordering process.

The extracted reordering rules were conducted by the survey of the English-Vietnamese parallel corpus with 131019 sentence pairs. We parse sentences into a dependency tree employing Stanford Parser [21] and extract the parent-child relationship including POS-tags of the parent node, the dependency label of the child node, the order of the child node with the parent node (position of child node relative to the parent node), the number of cases where the relations change position relative to the total number of relations. The reordering rule created from this process is a mapping from *T* to a

Algorithm 1. Extract rules with POS-tags and dependency label.

```
Input: $T$, $L = \{(dp_1, dp_2, ..., dp_n)\}$  // set of dependency labels, $n$: number of dependency labels of POS-tags $T$
                                                // self: head node's dependency label
Output: $R$ // set of reordering rules ($T$, $L'$, $W$)
        // $L'$: set dependency label head ordering, $W$: position of dependency label
init $R = \varnothing$
$L' = \{self\}$
repeat
    $c_1 := 0$, $c_2 := 0$ , $c_3 := 0$    // $c_1$: number of $dp_i$ in left-side of self, $c_2$: number of $dp_i$ keep position with self,
                                           $c_3$: number of dpi in right-side of self
    for each $dp_i \in (dp_1, dp_2, ..., dp_n)$ in $L$
        if $dp_1$ in left-side of $self$
            $c_1$ ++
        if $dp_2$ keep position with $self$
            $c_2$ ++
        if $dp_2$ in right-side of $self$
            $c_3$ ++
    end for
    $pos = \max\{c_1, c_2, c_3\}$
    if $pos = c_1$
            $L' = dp_i \rightarrow self$    //  insert $dp_i$ in the left-side off $self$
            $R = \{T, L', pos\}$;
    if $pos = c_2$
            $L' = L' \cup \{dp_i\}$    // insert $dp_2$ keep position with $self$
            $R = \{T, L', pos\}$;
    if $pos = c_3$
            $L' = self \rightarrow dp_i$   // insert $dp_2$ in right-side of $self$
            $R = \{T, L', pos\}$;
    $L = L \setminus \{dp_i\}$
until $L = \varnothing$
return $R$;
```





The head's Pos-tags	The child's denpendency label	Position
VB	aux, nsubj, self, advmod, advcl, discourse, neg, csubj, dep, preconj, expl, cop, appos	2
	rcmod, amod	1
	nsujpass	0
	possessive	-1
	Tmod, iobj, acomp, prt, parataxis, xcomp, ccomp, vmod, cc, conj, prep, punct, dobj	-2
NN	det, self, nsubj, cop, advmod, mark, aux, neg, num, csubj, predet, disscourse, advcl, expl, nsubjpass, tmod, preconj, auxpass, dobj, quantmod	2
	npadvmod	1
	prt	0
	number	-1
	xcomp, ccomp, parataxis, possessive, poss, dep, cc, nn, apposm, vmod, rcmod, conjm, prep, punct, amod	-2
JJS	self, nsubj, cop, advmod, mark, aux, advcl, csubj, poss, neg	2
	predet	1
	discourse	0
	dobj	-1
	possessive, xcomp, ccomp, amod, parataxis, appos, dep, cc, det, rcmod, conj, vmod, punct, prep	-2
TO	self, advmod, punct, preconj, neg, nsubj, mark, mwe, ccomp, discourse, dobj	2
	parataxis	1
	advcl	0
	xcomp	-1
	aux, tmod, prep, dep, cc, conj, pcomp, pobj	-2

Figure 6. Survey of the POS-tags type and dependency label's order in reordering word order.

set of tuples (L, W), containing all (T, L) combinations that can be found in the dataset:

- T is the part-of-speech (POS) tag of the head in a dependency parse tree node.
- L is a dependency label for a child node.
- W is a weight indicating the position value of that child node.

Traversing the dependency tree would start at the root as the head, and for every such head node, we start reordering it and its children based on our weighted rule above and continue the traversal recursively down to all its children, until we have reached all the leaf nodes in the tree.

3.1.2. Neural Network-Based Pre-Ordering for PBSMT

In this method, we would train a simple machine learning model which would be tasked to change the word order of source sentences to a corresponding order in the target language. English and Vietnamese are, respectively, the source language and target language in our discussion.

For example, when translating the English sentence:

That moment changed my life

to Vietnamese, we would like to reorder it as:

moment that changed life my.

This model will be tasked with reordering input sentences like the example above before delivering them to the translation model.

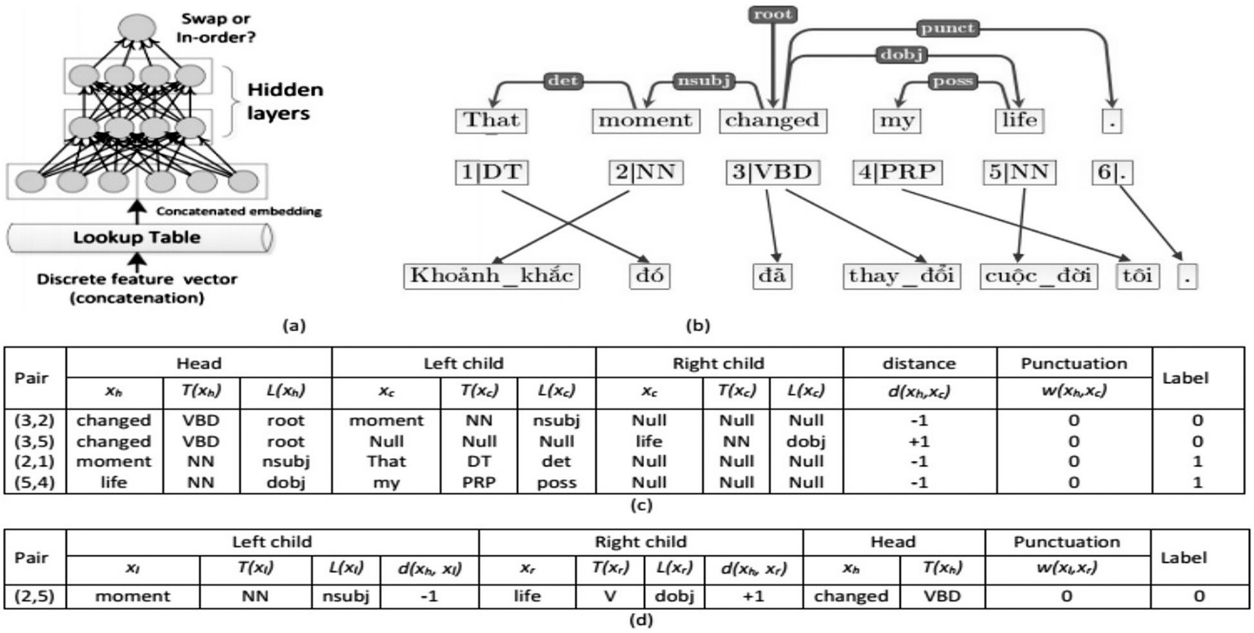
The classifier is built as a feed-forward neural network whose input layers contain features. By utilizing a lookup table, each feature is mapped into a continuous representative vector. The resulting vectors are concatenated and fed into a series of hidden layers by multiplying with weight matrices based on the rectified linear activation function $ReLU(x) = \max(0, x)$. Given that W is a weight vector, b is a bias value, x is the hidden layer transformed embedding vector. Inspired by [22], hidden layers and embedding layers for non-word features such as POS-tags, dependency labels, Boolean indicators are initialized by random uniform distribution while embedding for word features including x_h, x_c, x_l, and x_r are initialized by the dependency-driven embedding scheme [23]. This scheme works (in Figure 7) as a modified skip-gram model which predicts the context (surrounding words) of each given input word, then maps these words into similar continuous representative vectors if they have analogous surrounding words [24].

The prediction output δ is defined as:

$$z = W \cdot x + b \tag{1}$$

$$\delta(z) = \tanh(z) \tag{2}$$

$$L = \frac{1}{T}\sum_{i=1}^{T} y_i \log \widehat{y_i} + (1 - y_i)\log(1 - \widehat{y_i}). \tag{3}$$



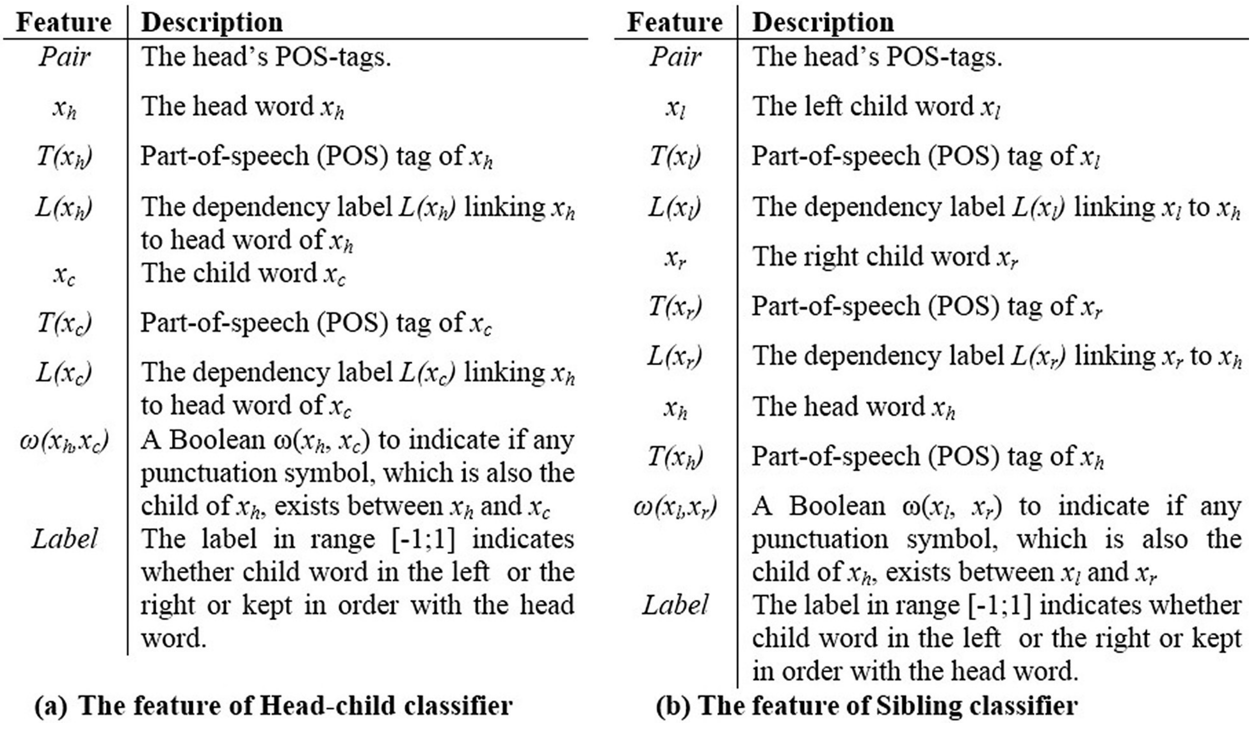
Pair	Head			Left child			Right child			distance	Punctuation	Label
	x_h	$T(x_h)$	$L(x_h)$	x_c	$T(x_c)$	$L(x_c)$	x_c	$T(x_c)$	$L(x_c)$	$d(x_h,x_c)$	$w(x_h,x_c)$	
(3,2)	changed	VBD	root	moment	NN	nsubj	Null	Null	Null	-1	0	0
(3,5)	changed	VBD	root	Null	Null	Null	life	NN	dobj	+1	0	0
(2,1)	moment	NN	nsubj	That	DT	det	Null	Null	Null	-1	0	1
(5,4)	life	NN	dobj	my	PRP	poss	Null	Null	Null	-1	0	1

(c)

Pair	Left child				Right child				Head		Punctuation	Label
	x_l	$T(x_l)$	$L(x_l)$	$d(x_h, x_l)$	x_r	$T(x_r)$	$L(x_r)$	$d(x_h, x_r)$	x_h	$T(x_h)$	$w(x_l,x_r)$	
(2,5)	moment	NN	nsubj	-1	life	V	dobj	+1	changed	VBD	0	0

(d)

Figure 7. Reordering Model: *(a)* neural network classifier architecture; *(b)* an aligned English-Vietnamese parallel corpus extracted training instances and features for *(c)* head-child classifier and *(d)* sibling classifier.



Feature	Description
Pair	The head's POS-tags.
x_h	The head word x_h
$T(x_h)$	Part-of-speech (POS) tag of x_h
$L(x_h)$	The dependency label $L(x_h)$ linking x_h to head word of x_h
x_c	The child word x_c
$T(x_c)$	Part-of-speech (POS) tag of x_c
$L(x_c)$	The dependency label $L(x_c)$ linking x_h to head word of x_c
$\omega(x_h,x_c)$	A Boolean $\omega(x_h, x_c)$ to indicate if any punctuation symbol, which is also the child of x_h, exists between x_h and x_c
Label	The label in range [-1;1] indicates whether child word in the left or the right or kept in order with the head word.

(a) The feature of Head-child classifier

Feature	Description
Pair	The head's POS-tags.
x_l	The left child word x_l
$T(x_l)$	Part-of-speech (POS) tag of x_l
$L(x_l)$	The dependency label $L(x_l)$ linking x_l to x_h
x_r	The right child word x_r
$T(x_r)$	Part-of-speech (POS) tag of x_r
$L(x_r)$	The dependency label $L(x_r)$ linking x_r to x_h
x_h	The head word x_h
$T(x_h)$	Part-of-speech (POS) tag of x_h
$\omega(x_l,x_r)$	A Boolean $\omega(x_l, x_r)$ to indicate if any punctuation symbol, which is also the child of x_h, exists between x_l and x_r
Label	The label in range [-1;1] indicates whether child word in the left or the right or kept in order with the head word.

(b) The feature of Sibling classifier

Figure 8. (a) The feature of head-child relation and *(b)* The feature of sibling relation.

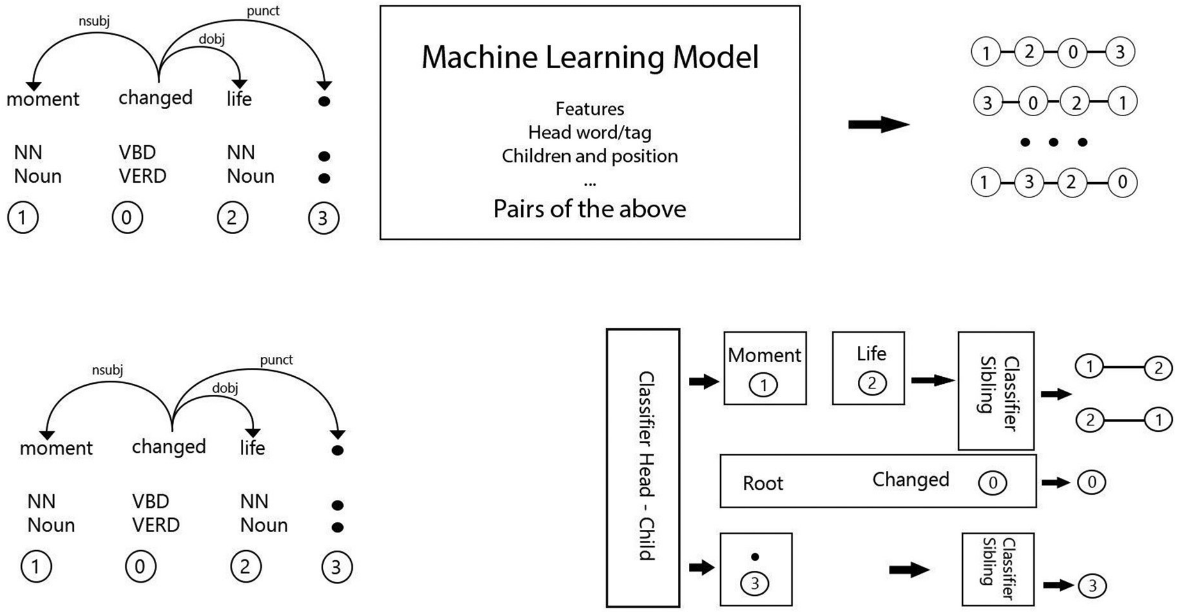


Figure 9. Classifier model for head-child relation and sibling relation.

The same procedure is conducted to define dependency information such as label, head word, and child word as context (Figure 8), in which similar heads and children are mapped into analogous continuous representative vectors. The training cases for the neural network classifiers are acquired from a word-aligned parallel corpus, from which we generate head-child or sibling relations from their corresponding order label. The order of these relations is either swapped or maintained based on the positions of their aligned target-side words. The neural network classifiers (Figure 9) are trained using backpropagation to minimize the binary cross-entropy loss function, since our final activation layer (tanh) is essentially a variant of a logistic unit and would benefit more from cross-entropy's robustness to gradient vanishing, as opposed to simpler loss function *e.g.*, mean-squared error.

Algorithm 2 extracts the features and builds models with dependency trees of source sentences and alignment pairs. Algorithm 3 predicts the order by considering head-child and sibling relations after it builds the new sentence from source-side dependency trees.

Algorithm 2. Build models.

```
Input: dependency trees of source sentences and alignment pairs
Output: two neural network classifier models:
        • PAC model (Head-child model)
        • SIB model (Sibling model)
for each head-child relation pair in dependency trees and alignment pairs of sentences do
    generate PAC_feature (head-child relation + label);
end for
for each sibling relation pair in dependency trees and alignment pairs of sentences do
    generate SIB_feature (sibling relation + label);
end for
build PAC_model from PAC_features;
build SIB_model from SIB_features;
return PAC, SIB;
```

Algorithm 3. Reordering.

```
Input: source sentence $S$, dependency tree $T$ of $S$;
Output: a new source sentence $S'$;
for each dependency tree of the source sentence do
    for each head-child relation in the tree $T$ do
        prediction head-child order from PAC model;
    end for
    for each sibling relation in the tree $T$ do
        prediction sibling order from SIB model;
    end for
end for
Build new sentence $S'$;
return $S'$;
```

The built model is expected to support the PB-SMT base model in the reordering process as another weighted reordering rule, leading to a more natural target word order compared to the vanilla version.

3.2. Artificial Noises

In Section 3.1, we have investigated the impact of differing word orders during translating and experimented with assimilating the source language's syntax to the target word order using pre-ordering. Alongside this technical goal, we study practical data augmenting techniques, supporting the reordering capacity of the models through varying target objectives, adding the secondary goal of removing noises or reordering broken input sequences. From these analyses, we propose a method to improve translation quality, by adding a denoising autoencoder objective to the concerning NMT system.

3.2.1. Denoising Autoencoder

An explorable aspect of translation is adapting different structures between language pairs and how to convert one syntactic structure to another. Specific neural sequence-to-sequence networks rely heavily on the input order (*e.g.*, Recurrent Neural Network – RNN using Long Short-Term Memory/Gate Recurrent Unit – LSTM/GRU cells), which often means that translation models would have trouble producing the correctly structured output as desired on low-resource data.

A way to mitigate such issues is to augment data and provide more training samples to help the model learn the specific target language structure. However, the direct and unmodified sequence used as the output may bias the model into repeating the words directly in a copied fashion; whereas a good model should learn to adapt and *generalize* the problem; hence the need to add noises to the input sequence so that the model can learn to differentiate between the correct and incorrect syntactic and grammatical cues of the corresponding target language. To this end, we further train our sequence-to-sequence model with monolingual data of the target language, generating an input sequence from the output by adding random noises (Figure 10).

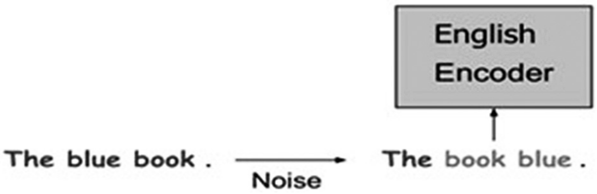


Figure 10. Injecting noise into an English (parent) source sentence.

The noise-generating functions are detailed below. To further aid the model in its generalizing capacity, our noises are generated differently for each epoch. The encoder is then tasked to rebuild the correct sentence structures of the target, thus learning the correct grammatical and syntactic structure of the language. As we can set the limit on the level of noise generation through the hyperparameters, we can be assured that our problem remains strictly on the level of denoising faulty sequences instead of ignoring the input altogether and attempting to memorize the training sequences.

The best theoretical input sequences for our denoising network would be the correct version of the monolingual sentences in the source language. However, acquiring such translation is infeasible in the case of low-resource languages; and even in the case of using an external translation model for this end, the correct re-alignment is often not available due to the limitation of supporting translation models and the tools needed to re-align the noisy generated source to the correct target sentence. Hence, by following the general autoencoder setup and by injecting artificial noise into a clean sentence to simulate the divergence between the source and target languages, we strengthen an aspect that a translating model has to adapt to without committing too much manual effort. We design different noise types, as follows, trying to mimic the different aspects that the NMT model will have to overcome during the translation process.

Inserting a word between original words. SMT translation by default outputs a target word for every source word; it can only be masked away by special tokens (*i.e.*, NULL) and not truly eliminated. However, in NMT, there are often cases where multiple source words combine into phrases corresponding to a single target word, or source words that should be removed altogether to make a cohesive output. For example, an English sentence "I came England" can be used as the input and the model should output the correct "I came from England" as the more natural English sentence (Figure 11).

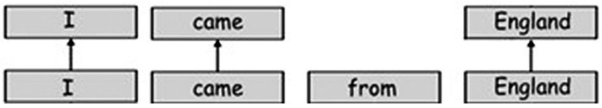


Figure 11. An example of adding an insertion noise.

By injecting noises to a clean target sentence, we attempt to imitate corresponding stop-words in the source language, forcing the model to ignore unimportant information. The noise adding process is set as:

1. For each position i, sample a probability $p_i \sim$ Uniform(0; 1).
2. If $p_i < p_{ins}$, sample a word from the most frequent V_{ins} target words and insert it before position i.

We use a value V_{ins} to limit the possible word insertion, as often the more frequent words tend to be prepositions or articles which contain relatively unimportant information while delivering the most disturbance to the sentence structure, such as the example above.

Deleting original words. Similarly, SMT does not exactly work in the reverse case: when a source word corresponds to a phrase represented as multiple target words, or when some target words should be created as padding from the nearby context, the model has to rely on a fertility value generated during translation to help with the former, or a special (NULL) token for the latter, both of which do not translate well to its corresponding NMT format. For example, the two English words "the not" must be "not" in English, and word translation should generate only the correct English words. This example is shown in Figure 12.

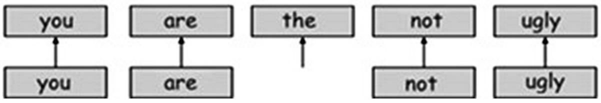


Figure 12. An example of adding a deletion noise.

Opposed to the insertion above, the deletion is to support generating fitting words from missing positions. Our method followed the work of Hill *et al.* [25]:

1. For each position i, sample a probability $p_i \sim$ Uniform(0; 1).
2. If $p_i < p_{del}$, drop the word in position i.

Permuting original word positions. Also, as the source and target language rarely share the exact word order even if they are from a close family of languages, there should be noises in the form of different positions to enhance robustness in both encoder and decoder. A common reordering problem of English- Vietnamese translation is illustrated in Figure 13.

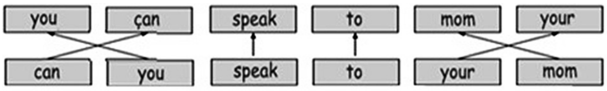


Figure 13. An example of adding the reordering noise.

For this method, we randomly choose positions to be swapped independently. We limit the maximum distance to move words using the method of Lample *et al.* [27]:

1. For each position i, sample an integer δ_i from [0; d_{per}].
2. Add δ_i to index i and sort the incremented indices $i + \delta_i$ in increasing order.

3. Rearrange the words to be in the new positions, to which their original indices have moved in Step 2.

This is a generalized version where the function only applies to neighboring words [25]. While this cannot generate word order like the source language, the noises introduced at this step can help the model generalize the target sentence structure. Insertion, deletion, and reordering noises were applied to the entire training data in each epoch with different random seeds, which created different noisy inputs from the same clean sentence during training. All three versions were included in our model and yielded acceptable results.

3.2.2. Back-Translation

Back-translation for NMT is particularly necessary for low-resource language pairs where bilingual data is scarce. The standard technique to address the scarcity is generating synthetic parallel data from target monolingual corpora via back-translation [28].

However, this approach works only if the generated source sentences are of sufficiently acceptable quality. Theoretically, back-translation is an enhancing technique that raises the capacity of the corresponding sequence-to-sequence model by strengthening the decoder's diversity in training data and forcing the encoder to learn to parse noisy input, both at the expense of training and inference power. While not exactly fitted to the task of enhancing reordering capacity, back-translation provides an interesting benchmark to the denoising encoder above, foregoing controlled noise generation in favor of one less model objective.

Provided the translation direction to be *S–T* and bilingual training data *D*(*x*, *y*) corresponding to that language pair with a monolingual corpus of *Y* in the target language *S*; the back-translation objective first requires a backward translation model *M'* with direction *T–S*. This backward translation model is often trained using the bilingual training data above. The model *M'*(*T–S*) would generate a faulty translation *X* from the monolingual corpus *Y* above, creating a pseudo-bilingual dataset *D'*(*X–Y*) to be added to the training data. Figure 14 gives an example using a back-translation method for English-Vietnamese machine translation.

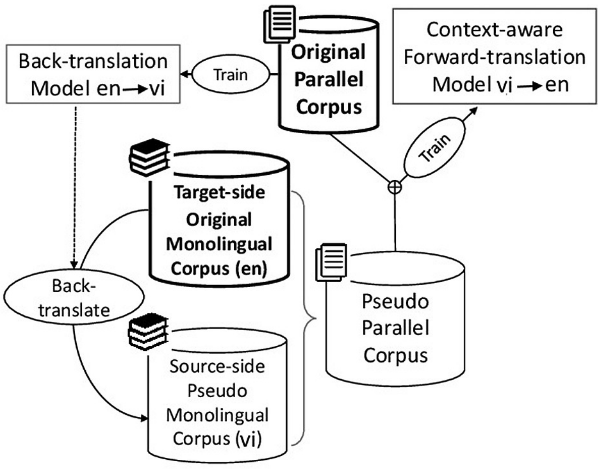

Back-translation Model en→vi
Train
Original Parallel Corpus
Context-aware Forward-translation Model vi→en
Train
Target-side Original Monolingual Corpus (en)
Back-translate
Pseudo Parallel Corpus
Source-side Pseudo Monolingual Corpus (vi)

Figure 14. An example using a back-translation method for English-Vietnamese machine translation.

The joined dataset *D* + *D'* is then used to train our main objective, the model *M*(*S–T*). Technically, the back-translation method can be used iteratively, as in generating the *M'*(*T–S*) model using a similar fashion with a *M''*(*S–T*) model to enhance the quality of the faulty translation *X*. However, as our goal is to introduce noises into our model to improve reordering capacity, we decided not to attempt this iterative technique.

4. Experiment

4.1. Data Set and Experimental Setup

For evaluation, an English-Vietnamese parallel corpus in the machine translation shared task of the IWSLT 2015 [29] was utilized, including 131,019 pairs for training, 1080 pairs for testing, and 1304 pairs for the development test set. Table 1 gives more statistical information about our corpora. Furthermore, some experiments with SMT Moses Decoder [30] and SRILM [31] were conducted. We trained a trigram language model using interpolate and discount smoothing with monolingual corpus, using GIZA++ [14] to build word alignment with the grow-diag-final-and algorithm before

Table 1. Corpus Statistics.

Corpus	Sentences pairs	Training set	Development set	Test set
General	133,403	131,019	1304	1080
			Vietnamese	English
Training	Sentences		131,019	
	Average Length		18.91	17.98
	Word		2,481,762	2,360,727
	Vocabulary		39071	54086
Development	Sentences		1304	
	Average Length		22.73	24.41
	Word		9092	8567
	Vocabulary		1537	1920
Test	Sentences		1080	
	Average Length		22.70	21.42
	Word		22,707	21,428
	Vocabulary		2882	3816

extracting the phrase table. We used the default reordering model in Moses Decoder and applied pre-processing to the source sentence with pre-ordering automatic rules.

We implemented the following.

- Before applying the pre-processing step for source sentences (English sentences), Stanford Parser [21] was used to parse source sentences.
- The pre-processing step was used in the training process and decoding time, using the SMT Moses decoder for decoding [24].
- For baseline NMT system, we used the Transformer architecture in NMT with 6 layers of encoder-decoder, internal vector size 512, the default absolute positional encoding; the batch size of 32 and dropout value of 0.1 on both embedding and feed-forward layers during training; the batch size of 8 and applying no unknown word replacement mechanism during inference.

4.2. Improving the Quality of PBSMT for English-Vietnamese Machine Translation

Using extracted rules. For the extracted rules, we used Stanford Parser [21] to parse source sentences and apply pre-processing steps for the source sentences (English sentences). We built a set of dependency-based rules for reordering words in English sentences according to Vietnamese word order, including noun phrase, adjectival and adverbial phrase, and preposition, based on typical differences in word order between English and Vietnamese. The English-Vietnamese parallel corpus and the dependency parser of English examples are used in the training model to automatically extract rules. Finally, we used these rules to reorder source sentences.

Using Auto-Rules by DPNN Classifier. The reordering decisions were made by two classifiers (head-child classifier and sibling classifier) where class labels correspond to the decision of 'swap' or 'not to swap'. We trained a separate classifier for each unique set of relationships. We did not learn explicit tree transformations rules in this method; instead, the classifiers learned to trade-off between a rich set of overlapping features. For the classification models needed for this method, we used neural network classification models. Starting from the root node, we applied them recursively top-down regarding the dependency tree. If the POS-tag of a node matched the left-hand side of the rule, the rule was applied, and the sentence's order was modified. We went over all the node's children and applied matching rules from the set of automatic rules to them.

Table 2 shows the BLEU scores of our experiments with manual rules using our methods and compared with the baseline of phrase-based SMT. These systems are compared with the baseline system (the state-of-the-art phrase-based system). The BLEU score of our best system increased by 0.67 points when pre-processing was used in both training and decoding, compared to the baseline system which is a powerful phrase-based SMT (integrating lexicalized reordering models). The result of applying transformation rules based on the dependency parse tree for the English-Vietnamese machine translation task was demonstrated above.

Table 2. Translation performance for the English-Vietnamese task using pre-ordering rules[1].

System	**BLEU (%)**	**Description**
Baseline	26.52	Phrase-based SMT
Baseline NMT	27.03	Neural Machine Translation with Transformer
Extracted rules	27.13	Phrase-based SMT using extracted rules
Auto-rules by DPNN Classifier	27.19	Phrase-based SMT using neural network classifier rules

4.3. Artificial Noises

For the denoising encoder objective, we filtered a 1M Vietnamese monolingual corpus downloaded from Wikipedia, removing sentences that have excessive complexity using a set of simple rules (special and different language characters, number of characters > 0.1, unknown words, and capitalization > 0.25), receiving 842k cleaned monolingual sentences. These sentences were used directly as output sequences of our model, while the input sequences were created by applying the noise-generating processes detailed above. The noisy sequences were recreated in each epoch to ensure maximum generalization capacity for our denoising objective. For the back-translating objective, we added 3M Vietnamese sentences found from CCMatrix [32] with the back-translating process detailed before.

For all three experiments, we used a normal Transformer architecture with 6 layers of encoder-decoder, the internal vector size of 512, the default absolute positional encoding; the batch size of 32, and dropout value of 0.1 on both embedding and feed-forward layers during training; the batch size of 8 and applying no unknown word replacement mechanism during inference. Learning rate was warm-started using a Noam Scheduler, with warm-up steps set to 4k for the baseline model and 12k for the two augmented models.

From the results presented in Table 3, it can be observed that both augmented versions

Table 3. Translation performance for the English-Vietnamese task using denoising autoencoder and back-translation[1].

System	**BLEU (%)**	**Description**
Baseline NMT	27.03	Neural Machine Translation with Transformer
Denoising autoencoder (+842k)	27.70	NMT with denoising autoencoder
Back-translation (+3M)	27.82	NMT with back-translation

[1] Result shown is the highest achieved score for each type of model.

- Baseline PBSMT and extracted rules used deterministic processes and don't have to be rerun.
- DPNN classifier chose the best performing classifier (best precision score on a tuning set).
- All baseline NMT models scored from 26.81 to 27.03
- All denoising autoencoder models scored from 27.33 to 27.70
- We only trained a single back-translation model.

Table 4. Examples of improvements made by denoising autoencoder and back-translation.

Source (English)	Reference and Translations (Vietnamese)
28: No apple has ever tasted the same .	**Reference**: Không có quả táo nào ngon được đến thế . **Baseline**: Không có quả táo nào cả . **Denoising**: Không có quả táo nào có vị như nhau cả . **Back-translation**: Không có quả táo nào có vị giống nhau .
396: 56 percent of all rape cases don &apos;t result .	**Reference**: 56 % vụ án hiếp dâm không được xử lý . **Baseline**: Hàng trăm trường hợp cưỡng hiếp không kết quả . **Denoising**: 56 % của tất cả các vụ cưỡng hiếp không gây ra kết quả gì . **Back-translation**: 56 % tất cả các trường hợp hiếp dâm không dẫn đến .
728: So I signed the discharge papers , and an ambulance came , paramedics came to take her home .	**Reference**: Thế là tôi kí giấy tờ xuất viện , và xe cứu thương tới , y sĩ đưa bà về nhà . **Baseline**: Thế là tôi ký báo giấy , và một chiếc xe cứu thương đến , <unk> đến để lấy nhà . **Denoising**: Vì thế tôi đã ký giấy khai thác , và xe cấp cứu đến , nhân viên cứu thương đến mang cô về nhà . **Back-translation**: Vì vậy tôi đã ký giấy cứu trợ , và một xe cứu thương đến , nhân viên y tế đến nhà cô ấy .
1107: But then what happened was the individuals worked out , of course , tricks of communicating .	**Reference**: Nhưng sau đó điều xảy ra là các cá thể hình thành nên đương nhiên rồi , những mánh khoé trong việc trao đổi thông tin . **Baseline**: Nhưng rồi những gì xảy ra là những cá nhân đã thành công , tất nhiên , là những trò lừa giao tiếp . **Denoising**: Nhưng rồi những gì xảy ra là những cá nhân xuất hiện , tất nhiên , là những trò lừa bịp của việc giao tiếp . **Back-translation**: Nhưng sau đó những gì đã xảy ra là những cá nhân đã tìm ra , tất nhiên , những thủ thuật giao tiếp .

performed better than the baseline version by approximately 0.5 BLEU, with the back-translation method receiving a slightly better score. In practice, this difference between augmented versions does not seem to exhibit any specific change, which suggests that both techniques ended up improving the same capacity of the translation model; hence, it might be better to use the denoising encoder variant in most cases as it traded a negligible amount of translation accuracy for a considerable improvement in training and inference capacity. Some similar improvements between these two methods are shown in Table 4, illustrating that both attempt to enhance translations with more information from the source side.

5. Discussion

Our work focused solely on linguistic phenomena specialized for the English-Vietnamese language pair translation. We cannot extract syntax-based rules from NMT systems about these languages, nor are we capable of altering their linguistic capacity without requiring more parallel data, which is quite costly to build, especially on rarer languages. Therefore, our alternative is to apply pre-ordering rules to PBSMT to help solve this problem, relying on their proven adequacy to ensure translation integrity. In parallel, we also explore ways to apply reordering noises to NMT systems with the same goal as an additional objective and a supporting benchmark. To support these objectives, we analyzed grammatical structures, morphology, and the larger linguistic phenomena in the case of Vietnamese sentences translating to the English language, focusing on special divergences, such as pre-modifiers.

Overall, our work represented substantial improvements in translation performance in both directions. For the PBSMT reordering rule, we based it on the respective dependency grammars, in which there are approximately 50 grammatical relations in English and 27 ones in Vietnamese, along with identified word order differences between English and Vietnamese to create a set of the reordering rules. For the NMT based system, we utilized monolingual data and applied well-known noise-generating concepts and proven methods, proving that the goal of improving translation quality correlates to the improvement in reordering capacity.

We compared the results of phrase-based SMT, phrase-based SMT with pre-ordering rules, neural machine translation, NMT with denoising autoencoder, and back-translation. We believe that focusing on the reordering aspect of the machine translation process can improve the quality of both phrase-based SMT systems and NMT, especially with low-resource language pairs.

6. Conclusion

In this paper, we proposed methods to improve the quality of machine translation by denoising autoencoders in NMT systems and pre-ordering in PBSMT where the neural-based and phrase-based have become dominant among current machine translation methods. We present several variants of an autoencoder and corresponding data to aid the process of reordering target sentences to support and compare our options. The experimental results show that we can acquire an improvement in translation accuracy in low-resource domains using a simple process of adding noise to synthesize training data.

Our proposed models can be efficiently trained with little or no changes in the implementation, with a quick process to create noise from existing monolingual data. The provided analyses help to better learn linguistics phenomena for translation purposes. In effect, the denoising autoencoder has a result comparable to the one utilizing samples from the opposite translation direction (back-translation) without the same inference cost. This approach has a lot of potential to improve the quality of machine translation for the reordering problem, a very important aspect of the translation task in general.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments that greatly contributed to improving the quality of the paper. This work is supported by the Ministry of Science and Technology of Vietnam under Program KC 4.0, No. KC-4.0.12/19-25.

References

[1] K. Papineni *et al.*, "Bleu: A Method for Automatic Evaluation of Machine Translation", in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, pp. 311–318.

[2] M. Collins *et al.*, "Clause Restructuring for Statistical Machine Translation", in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, 2005.

[3] A. Ramanathan *et al.*, "Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation", in *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, 2008.

[4] J. Navratil *et al.*, "A Comparison of Syntactic Reordering Methods for English-German Machine Translation", in *Proceedings of COLING 2012, The 24th International Conference on Computational Linguistics, COLING*, 2012.

[5] R. Chatterjee *et al.*, "Super Tag Based Preordering in Machine Translation", in *Proceedings of the 11th International Conference on Natural Language Processing, ICON*, 2014.

[6] E. M. Ponti *et al.*, "Isomorphic Transfer of Syntactic Structures in Cross-Lingual nlp", in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL*, 2018.

[7] Y. Kawara *et al.*, "Recursive Neural Network-Based Preordering for English-to-Japanese Machine Translation", in *Proceedings of ACL 2018, Student Research Workshop*, 2018, pp. 21–27. http://dx.doi.org/10.18653/v1/P18-3004

[8] T. Q. Nguyen and D. Chiang, "Transfer Learning Across Low-Resource, Related Languages for Neural Machine Translation", in *Proceedings of the 8th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), IJCNLP*, 2017.

[9] J. Lee *et al.*, "Fully Character-Level Neural Machine Translation without Explicit Segmentation", *Transactions of the Association for Computational Linguistics*, 2017.

[10] J. Gu *et al.*, "Universal Neural Machine Translation for Extremely Low Resource Languages", in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), NAACL*, 2018.

[11] K. Knight and Y. Al-Onaizan, "Translation with Finite-State Devices", *Lecture Notes in Artificial Intelligence*, Springer-Verlag, vol. 1529, pp. 421–437, 1998.

[12] S. Kumar and W. Byrne, "A Weighted Finite-State Transducer Implementation of the Alignment Template Model for Statistical Machine Translation", in *Proceedings of the Human Language Technology Conf. NAACL*, 2003, pp. 142–149.

[13] D. Chiang, "A Hierarchical Phrase-Based Model for Statistical Machine Translation", in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, 2005, pp. 263–270.

[14] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models", *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[15] Y. Zhang *et al.*, "Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation", in *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, 2007, pp. 1–8.

[16] P. Xu *et al.*, "Using a Dependency Parser to Improve SMT for Subject-Object-Verb Languages", in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics*, 2009, pp. 245–253.

[17] D. Genzel, "Automatically Learning Source-Side Reordering Rules for Large Scale Machine Translation", in *Proceedings of the 23rd International Conference on Computational Linguistics, COLING'10*, 2010, pp. 376–384.

[18] U. Lerner and S. Petrov, "Source-Side Classifier Preordering for Machine Translation", in *Proceedings of the EMNLP*, 2013, pp. 513–523.

[19] D. Bahdanau *et al.*, "Neural Machine Translation by Jointly Learning to Align and Translate", in *Proceedings of the 2015 International Conference on Learning Representations (ICLR)*, 2015.

[20] A. Vaswani *et al.*, "Attention is all You Need", in *Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017, pp. 5998–6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need

[21] D. Cer *et al.*, "Parsing to Stanford Dependencies: Trade-Offs Between Speed and Accuracy", in *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, 2010.

[22] C. Hadiwinoto and H. T. Ng, "A Dependency-Based Neural Reordering Model for Statistical Machine Translation", arXiv preprint arXiv:1702.04510, 2017.

[23] M. Bansal *et al.*, "Tailoring Continuous Word Representations for Dependency Parsing", in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 809–815.

[24] T. Mikolov *et al.*, "Efficient Estimation of Word Representations in Vector Space", in *Proceedings of the International Conference on Learning Representations (ICLR) Workshop, CoRR abs/1301.3781*, 2013, pp. 746–751.

[25] F. Hill *et al.*, "Learning Distributed Representations of Sentences from Unlabelled Data", in *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, 2016, pp. 1367–1377.

[26] Y. Kim *et al.*, "Improving Unsupervised Word-by-Word Translation with the Language Model and Denoising Autoencoder", in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 862–868.

[27] G. Lample *et al.*, "Unsupervised Machine Translation Using Monolingual Corpora Only", in *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*, 2018.

[28] R. Sennrich *et al.*, "Improving Neural Machine Translation Models with Monolingual Data", in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 86–96, 2016.

[29] Cettolo, Mauro, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni and Marcello Federico. The international workshop on spoken language translation (iwslt). http://workshop2015.iwslt.org/index.php, IWSLT (2015).

[30] P. Koehn *et al.*, "Open Source Toolkit for Statistical Machine Translation", in *Proceedings of ACL, Demonstration Session*, 2007.

[31] A. Stolcke, "Srilm – An Extensible Language Modeling Toolkit", in *Proceedings of International Conference on Spoken Language Processing*, 2002, pp. 901–904.

[32] H. Schwenk *et al.*, "CCMatrix: Mining Billions of High-Quality Parallel Sentences on the WEB", CoRR, 2019 http://arxiv.org/abs/1911.04944

[33] Z. Xie *et al.*, "Noising and Denoising Natural Language: Diverse Back-Translation for Grammar Correction", in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL*, 2018. https://aclanthology.org/N18-1057

[34] Y. Kim *et al.*, "Effective Cross-Lingual Transfer of Neural Machine Translation Models without Shared Vocabularies", pp. 1246–1257, 2019. http://dx.doi.org/10.18653/v1/P19-1120

Received: September 2021
Revised: February 2022
Accepted: February 2022

Contact addresses:

Tran Hong-Viet
University of Economic and Technical Industries
Hanoi
Vietnam
e-mail: thviet79@gmail.com

Nguyen Van-Vinh
University of Engineering Technology
Vietnamese National University
Hanoi
Vietnam
e-mail: vinhnv@vnu.edu.vn

Nguyen Hoang-Quan
University of Engineering Technology
Vietnamese National University
Hanoi
Vietnam
e-mail: quanfm94@gmail.com

Tran Hong-Viet received his PhD degree in computer science in 2019 from the University of Technology, Hanoi National University (UET-VNU). His research interests include machine translation, machine learning, deep learning, natural language processing, computer vision, health care, and artificial intelligence.

Nguyen Van-Vinh working at the Department of Computer Science, University of Technology, Vietnam National University, Hanoi, received a PhD degree in computer science from the Japan Advanced Institute of Science and Technology (JAIST) in 2009. His research directions are machine translation, natural language processing, artificial intelligence, and machine learning.

Nguyen Hoang-Quan Nguyen Hoang Quan, working at the Department of Computer Science, University of Technology, Hanoi National University, graduated with a bachelor's degree in computer science from the University of Technology (UET-VNU) in 2018. He is currently studying for a master's degree in computer science at the University of Technology, Vietnam National University, Hanoi. His research directions are machine translation, natural language processing, artificial intelligence, and machine learning.