Improving Data Augmentation for Low-Resource NMT Guided by POS-Tagging and Paraphrase Embedding

MIERADILIJIANG MAIMAITI, State Key Laboratory of Intelligent Technology and Systems, Institute for Artificial Intelligence, Tsinghua University, Beijing National Research Center for Information Science and Technology (BNRist), Department of Computer Science and Technology, Tsinghua UniversityChina
YANG LIU[*], State Key Laboratory of Intelligent Technology and Systems, Institute for Artificial Intelligence, Tsinghua University, Beijing National Research Center for Information Science and Technology (BNRist), Department of Computer Science and Technology, Tsinghua University, China
HUANBO LUAN, State Key Laboratory of Intelligent Technology and Systems, Institute for Artificial Intelligence, Tsinghua University, Beijing National Research Center for Information Science and Technology (BNRist), Department of Computer Science and Technology, Tsinghua University, China
ZEGAO PAN, School of Software, Xinjiang University, Urumqi, China, China
MAOSONG SUN[†], State Key Laboratory of Intelligent Technology and Systems, Institute for Artificial Intelligence, Tsinghua University, Beijing National Research Center for Information Science and Technology (BNRist), Department of Computer Science and Technology, Tsinghua University, China

Data augmentation is an approach for several text generation tasks. Generally, in the machine translation paradigm, mainly in low-resource language scenarios, many data augmentation methods have been proposed. The most used approaches for generating pseudo data mainly lay in word omission, random sampling, or replacing some words in the text. However, previous methods barely guarantee the quality of augmented data. In this work, we try to build the data by using paraphrase embedding and POS-Tagging. Namely, we generate the fake monolingual corpus by replacing the main four POS-Tagging labels, such as noun, adjective, adverb, and verb based on both the paraphrase table and their similarity. We select the bigger corpus size of the paraphrase table with word-level and obtain the word embedding of each word in the table, then calculate the cosine similarity between these words and tagged words in the original sequence. In addition, we exploit the ranking algorithm to choose highly similar words to reduce semantic errors and leverage the POS-Tagging replacement to mitigate syntactic error to some extent. Experimental results show that our augmentation method consistently outperforms all the previous SOTA methods on the low-resource language pairs in 7 language pairs from 4 corpora by 1.16 ∼ 2.39 BLEU points.

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

Additional Key Words and Phrases: Artificial Intelligence; Natural Language Processing; Neural Network; Machine Translation; Low-Resource Languages; Data Augmentation; POS-Tagging; Paraphrase

1 INTRODUCTION

Recently, end-to-end architecture has been a commonly used neural network (NN) in sequence generation tasks. Neural machine translation (NMT), which uses neural networks to model the translation process of natural languages in an end-to-end manner, has attracted intense attention from the community [Bahdanau et al. 2015; Sutskever et al. 2014; Vaswani et al. 2017; Wu et al. 2016]. Excelling in learning representations and capturing long-distance dependencies by exploiting gating [Cho et al. 2014; Hochreiter and Schmidhuber 1997], attention mechanisms [Bahdanau et al. 2015], pre-training [Devlin et al. 2019], ensembling [Wang et al. 2020], regularization [Zhang et al. 2019], and weighting [Wang et al. 2018b] methods, NMT has shown significant advantages over conventional statistical machine translation (SMT) [Brown et al. 1993; Chiang 2005; Koehn

[*] Also with the Beijing Academy of Artificial Intelligence, Beijing Advanced Innovation Center for Language Resources.
[†] Also with Jiangsu Collaborative Innovation Center for Language Competence, Jiangsu, China.

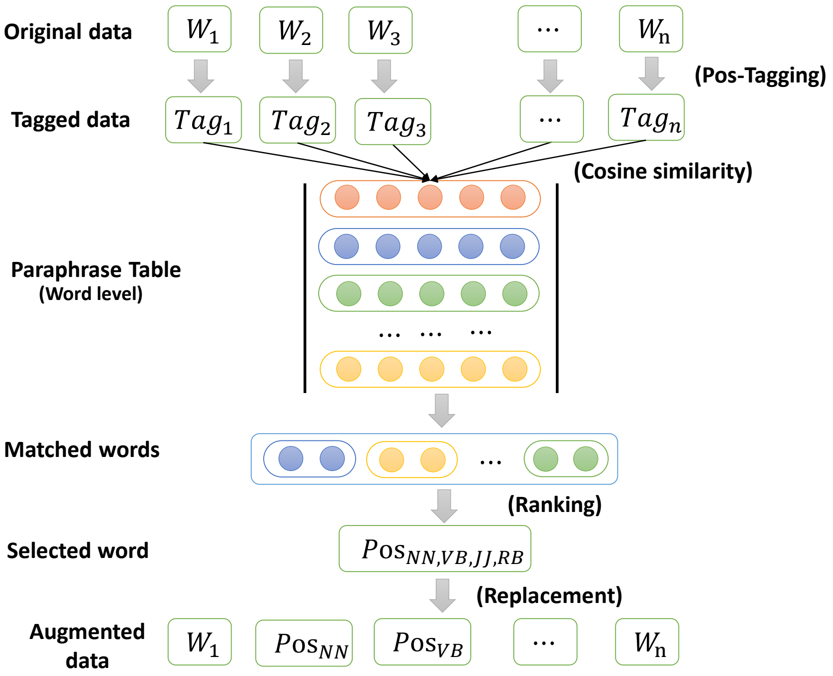


Fig. 1. The main architecture for data augmentation guided by part-of-speech tagging and paraphrase embeddings, where "POS_{NN}", "POS_{VB}", "POS_{JJ}" and "POS_{RB}" denote mainly replaced POS-Tagging labels noun, verb, adjective and adverb, respectively.

et al. 2003] for a number of natural languages [Junczys-Dowmunt et al. 2016]. The remarkable performance enhances traditional SMT on several languages, making NMT an appealing approach to the real-world machine translation community.

However, NMT needs large numbers of parallel corpora, and therefore it usually learns unsatisfactorily on low-resource languages (LRLs) [Chen et al. 2017]. It is easy to gain large amounts of parallel corpora in high-resource languages (HRLs). The high agglutinations of LRLs bring some challenges for training the NMT model on LRLs [Vania and Lopez 2017]. In the machine translation (MT) scenario, morphologically rich languages always have a data scarcity problem [Chen et al. 2018]. Therefore, we focus on this issue in our proposed model. Data augmentation (DA) is an essential method, as it builds extra data to enhance the accuracy of deep learning approaches. Meanwhile, these generated corpora may have lower quality than the original real training data. So NMT considers the model quality as a tough problem, and it roughly depends on the accessibility of huge amounts of parallel corpus. Moreover, it is shown to remarkably underperform SMT since the neural network tends to learn defectively on LRLs.

Translating from LRLs is one of the biggest challenging issues. DA has been commonly used in many fields. In computer vision, the major DA approaches include some transformations like resizing, rotating, cropping, and flipping [Cubuk et al. 2019]. Furthermore, similar conversion methods [Xie et al. 2017] have also been broadly leveraged in NLP tasks. However, due to the text characteristics, the common random transformation usually brings trivial changes and even makes some semantic or syntactic mistakes. Contextual augmentation [Wu et al. 2019] can be regarded as a novel method, which exploits the language model (LM) to replace some words with each other. The drawback of this method is that several unseen samples have high variations and they require a lot of time to sample. Additionally, since the minor revision of sequence can result in extreme changes in their semantics, it is problematic to seek simple correspondence for some tough NLP tasks like MT. Because of such complexity, the previous literature in DA for NMT is rather insufficient. Artetxe et al. [2018] swap the words in the sequence randomly. Iyyer et al. [2015] drop some words arbitrarily in a sentence to help the training process of NMT via learning encoders. Xie et al. [2017] show that data noising is an effective trick for NMT and they implement their idea by instituting the words with a placeholder. Fadaee et al. [2017] also gain improvements in NMT by alternating the words in the target with rare words and revise the corresponding words in the source. Kobayashi [2017] present a method which leverages prior knowledge by taking advantage of the bi-directional language model to replace word tokens. Sugiyama and Yoshinaga [2019] exploit contexts around the source sentence, and have so far explored context-aware NMT to build some pseudo parallel corpus.

In this paper, we derive a rather straightforward but effective generation method for data augmentation in NMT. We take DA as a sampling method and introduce a novel DA trick for MT in LRLs, and we believe the POS-Tagging and paraphrase embedding are more efficient than other approaches which randomly sample some words among the original texts. Additionally, in this experiment, we do not design any evaluation model, because the tagged words guarantee a syntactic issue, and embedding similarity also mitigates the semantic errors. Precisely, as depicted in Figure 1, our method can be decomposed into three steps. Firstly, we tag each of the words in the sequence using the NLTK toolkit (for English). In this step, we only label four types of frequent tags, such as name, adjective, adverb, and verb. Secondly, we select the bigger corpus size of the paraphrase table with word-level and obtain the word embedding of each word in the table, then we calculate the cosine similarity between these words and tagged words in the original sequence. Thirdly, we achieve some matched words using word similarity and select the highly close words by exploiting the ranking method. Finally, we replace the words with selected words to augment the new pseudo sequence. Our contributions are as follows:

- We apply the POS-Tagging to reduce the syntactic errors from generated data during data augmentation.
- We mitigate the semantic errors based on the paraphrase table by employing the word similarity.
- We reserve the higher quality of sentences with ranking before word replacement.
- Our augmentation method is model transparent and language independent.

2 BACKGROUND

2.1 Neural Machine Translation

Generally, we can regard X as a source language sentence and Y as a target language sentence. Given a source sentence $\mathbf{x} = x_1, \ldots, x_i, \ldots, x_I$ and a target sentence $\mathbf{y} = y_1, \ldots, y_j, \ldots, y_J$, standard NMT models [Bahdanau et al. 2015; Sutskever et al. 2014; Vaswani et al. 2017] usually factorize the sentence-level translation probability as a product of word-level probabilities:

$$P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) = \prod_{j=1}^{J} P(y_j|\mathbf{x}, \mathbf{y}_{<j};\boldsymbol{\theta}), \tag{1}$$

where $\boldsymbol{\theta}$ is a set of model parameters, and $\mathbf{y}_{<j}$ is a partial translation.

Let $\langle \mathbf{X}, \mathbf{Y} \rangle = \{\langle \mathbf{x}^{(n)}, \mathbf{y}^{(n)} \rangle\}_{n=1}^{N}$ be a training corpus. The log-likelihood of the training parallel data is maximized by the standard training objective:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left\{ \sum_{n=1}^{N} \log P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}; \boldsymbol{\theta}) \right\}. \tag{2}$$

The translation decision rule for unseen source sentence $\mathbf{x}$ given learned model parameters $\hat{\boldsymbol{\theta}}$ is given by

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \left\{ P(\mathbf{y} | \mathbf{x}; \hat{\boldsymbol{\theta}}) \right\}. \tag{3}$$

Meanwhile, calculating the highest probability $\hat{\mathbf{y}} = \hat{y}_1, \ldots, \hat{y}_j, \ldots, \hat{y}_J$ of the target sentence can be separated at the word level:

$$\hat{y}_j = \underset{y}{\operatorname{argmax}} \left\{ P(y | \mathbf{x}, \hat{\mathbf{y}}_{<j}; \hat{\boldsymbol{\theta}}) \right\}. \tag{4}$$

2.2 Transformer

In the NMT community, The transformer has become one of the most well-known neural machine translation architectures. The main architecture of the transformer is also composed of two parts such as encoder and decoder. The encoder includes stacks of N identical layers. Each of them consists of two sub-layers, i.e., the multi-head self-attention layer and the simple feed-forward network. Generally, the self-attention sub-layer first runs to interact information between various words. It employs h attention heads, which allows the model to jointly attend to capturing features from different representation subspaces and different positions. A single attention head is calculated on a query Q, a key K and a value V:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}} \right) V. \tag{5}$$

Consequently, the self-attention layer combines the h different representations of (Q, K, V) and maps the concatenation to take advantage of a feed-forward layer to generate the output:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W^O, \tag{6}$$

where each head in MultiHead is noted as below:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V). \tag{7}$$

Furthermore, the feed-forward sub-layer connected behind the self-attention mechanism runs as:

$$\text{FFN}(x) = \max(0, xW_1 + b_1) W_2 + b_2. \tag{8}$$

Between each of the two sub-layers, we also exploit a residual connection [He et al. 2016] and follow the idea of layer normalization [Ba et al. 2016]. Consequently, the output which is generated from each sub-layer is $\text{LayerNorm}(x + \text{Sublayer}(x))$, where $\text{Sublayer}(x)$ is a representation implemented by the sub-layer itself.

2.3 Data Augmentation

In the text generation paradigm, the data augmentation method has been used widely, such as computer vision [Cubuk et al. 2019], dialogue generation [Ke et al. 2019], MT [Fadaee et al. 2017], and other NLP tasks [Xie et al. 2017]. Therefore, data augmentation is still a ubiquitous and pervasive trick for low-resource languages in neural machine translation, which generates some pseudo data D_g using some common methods from the original data D_o. As illustrated in Figure 2, common

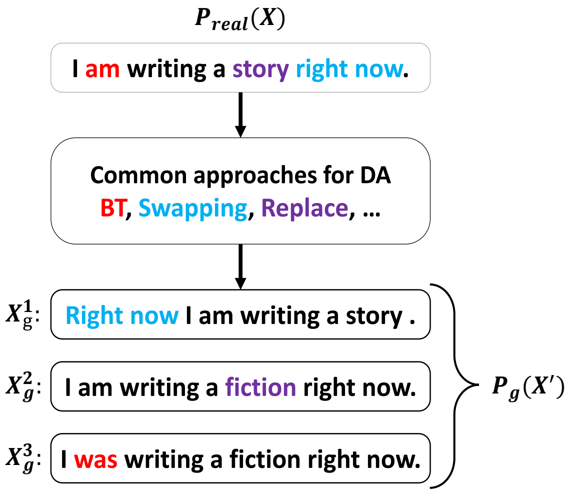

$P_{real}(X)$
I am writing a story right now.
Common approaches for DA
BT, Swapping, Replace, ...
X_g^1: Right now I am writing a story .
X_g^2: I am writing a fiction right now.
X_g^3: I was writing a fiction right now.
$P_g(X')$

Fig. 2. The general architecture of DA for NMT, where $P_{real}(X)$ denotes the original real data distribution, $P_g(X^{'})$ represents the generated fake pseudo data distribution, and "X_g^1", "X_g^2" and "x_g^3" refer to augmented data generated from real data using common approaches such as swapping, replacing and Bt.

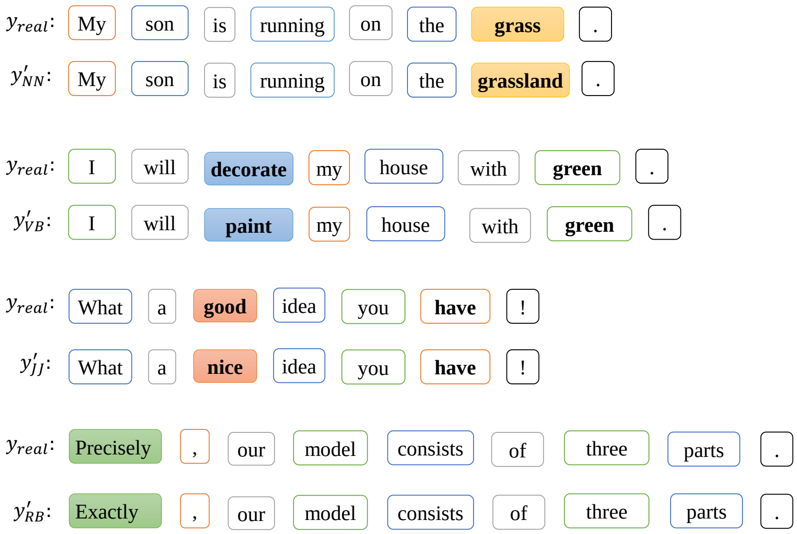
Algorithm 1: Data Augmentation with POS-Tagging and Paraphrase

```
Input: Original data $D_o = \{\langle \mathbf{x}^{(n)}, \mathbf{y}^{(n)} \rangle\}_{n=1}^{N}$.
Output: Generated data $D_g$.
/* Data preparation step.                                                */
Prepare the larger size of word level paraphrase table $Paraphrase_i = \{w^{(i)}\}_{n=1}^{I}$.
/* Use Stanford POS-Tagger.                                              */
Exploit the POS-Tagger to label tags noun, verb, adjective, and adverb in the target side from
  the original data $D_o$.
Calculate the cosine similarity between the four labels from $D_o$ and the words $w_i$ from
  $Paraphrase_i$ using pre-trained embedding.
/* Based on the similarity to achieve the similar word.                  */
Choose the highest similar words from word buffers by ranking.
Replace the original POS-Tagging with chosen words and generate the new data $D_g$.
/* Obtain the final augmented data.                                      */
Get the new training data $D_o \cup D_g$ with $4 * N$ sentence pairs. ;      /* for 4 tags */
```

DA techniques used with NMT include: swapping two words randomly [Lample et al. 2018], using back-translation (Bt) [Sennrich et al. 2016a] and replacing words [Fadaee et al. 2017] with different words among a given sequence. Generally, for a real sequence $X = w_1, w_2, w_3, \ldots, w_n$ which is composed of n words, the previous methods encourage the augmentation model to build the new sequence $X_g = \hat{w}_1, \hat{w}_2, \hat{w}_3, \ldots, \hat{w}_n$ by leveraging random manner.

3 METHODOLOGY

Neural machine translation demands massive corpora, and it has poor performance due to the data scarcity in LRLs. Therefore, DA attracts more attention than other commonly used techniques for LRLs. However, the generated data after DA can not be guaranteed to have no syntax or semantic



y_{real}: My son is running on the **grass** .

y'_{NN}: My son is running on the **grassland** .

y_{real}: I will **decorate** my house with **green** .

y'_{VB}: I will **paint** my house with **green** .

y_{real}: What a **good** idea you **have** !

y'_{JJ}: What a **nice** idea you **have** !

y_{real}: Precisely , our model consists of three parts .

y'_{RB}: Exactly , our model consists of three parts .

Fig. 3. The process of target augmentation by leveraging POS-Tagging labels, where "y_{real}" represents the original real target data, and "y_{NN}", "y_{VB}", "y_{JJ}" and "y_{RB}" denote the augmented data by replacing the noun, verb, adjective and adverb, respectively.

errors. Another motivation is to achieve better translation results using the DA methods without considering designing an independent model for avoiding grammar errors. Therefore it is crucial to analyze the perplexity and adequacy of the augmented data; i.e., we need to build high quality pseudo-data by exploiting simple architecture. To solve this problem, we design a straightforward and effective architecture and propose the method to augment the original real data by replacing the words with different POS-Tagging based on the paraphrase corpus in various granularity. As illustrated in Figure 3, we regard the DA task as follows:

Given a parallel corpus $D_o = \{\langle \mathbf{x}^{(n)}, \mathbf{y}^{(n)} \rangle\}_{n=1}^{N}$, we select the target side monolingual data ($y_{real} = w_1, w_2, ..., w_i, ..., w_n$), and in this work we take the English as our target train set. As shown in Algorithm 1, firstly we need to prepare a larger size of the word-level paraphrase table $Paraphrase_i = \{w^{(i)}\}_{n=1}^{I}$. Secondly, label each of the words w_i in the D_o using the NLTK toolkit to achieve their corresponding POS-Tags ($POS_{NN}, POS_{VB}, POS_{JJ}$ and POS_{RB}). Then, exploit the pre-trained word-to-vector model to obtain embeddings of tagged words and the words in paraphrase table $Paraphrase_i$. With respect to the cosine similarities between these words, we can obtain a bunch of matched words. Thirdly, employ the ranking on cosine similarity to select the highly similar word with tagged words. Finally, we build the newly generated data D_g by replacing the selected words which are achieved from the previous step.

Precisely, if we need to replace the word "accurately" whose tag is an adverb in the current sequence ($y_{real} = Accurately, w_2, ..., w_i, ..., w_n$), after labeling we will obtain the tagged sequence ($y_{real} = w_{POS_{RB}}, w_2, ..., w_i, ..., w_n$) , where $w_{POS_{RB}}$ is the adverb and $\hat{w}_i$ is a word from a $Paraphrase_i$. We may gain a bunch of matched words "exactly, precisely, obviously, apparently, evidently ..." with different probabilities by calculating cosine similarity. We can find the highly similar word (e.g. "exactly"), then replace them with each other to build D_g. The replacement of different parts of speech will achieve the corresponding sequence word by word sequentially. Besides, in the replacing step we only substitute a word in time, instead of making replacement more than a word each time.

Table 1. Characteristics of our corpora.

Language Pairs	Train	Dev.	Test	Source			Target		
				Vocab.	Token.	Avglen.	Vocab.	Token.	Avglen.
Az → En	21.2K	1.0K	1.0K	24.6K	3.1M	14.50	18.9K	4.0M	18.77
Hi → En	182.0K	1.0K	1.0K	13.3K	7.3M	39.97	20.1K	5.0M	27.09
Uz → En	134.6K	1.0K	1.0K	25.2K	2.2M	15.97	20.1K	2.5M	18.12
Tr → En	141.9K	1.0K	1.0K	86.0K	0.8M	5.85	54.4K	1.0M	6.86
De → En	160K	7.3K	6.8K	113.5K	3.1M	19.35	53.3K	3.3M	20.44
Vi → En	140.5K	1.6K	1.3K	25.3K	3.5M	24.64	48.64K	2.9M	20.10
Zh → En	1.0M	0.9K	0.9K	191.3K	22.7K	22.68	97.8k	28.4M	28.42

“Dev." indicates the validation set. “Vocab.” and “Token.” denote vocabulary (non-repeated) and all tokens (including repeated), respectively. “Avglen.” represents the average length of sentences.

Table 2. The statistics of POS-Tagging labels in the corpora.

POS_{Labels}	Az - En	DE - En	Hi - En	Vi - En	Tr - En	Uz - En	Zh - En
POS_{NN}	779.2K	798.7K	961.0K	785.5K	4.1M	498.2K	8.0M
POS_{VB}	865.3K	62.2K	1.1M	513.5K	4.8M	537.3K	4.0M
POS_{JJ}	192.6K	235.3K	246.1K	207.2K	1.0M	122.9K	2.8M
POS_{RB}	192.4K	188.5K	240.1K	150.1K	1.0M	118.4K	912.4K
POS_{ALL}	2.0M	1.3M	2.5M	1.7M	11M	1.3M	15.6M

The subscripts “NN", “VB", “JJ", “RB" and “ALL" represent the POS-Tagging of a noun, verb, adjective, adverb and the summation of all POS-Tagging labels in the corpus, respectively.

4 EXPERIMENTS

4.1 Setup

Data Preparation. The source side of the language pairs which was used in our main experiment is composed of Azerbaijan (Az), Hindi (Hi), Uzbek (Uz), and Turkish (Tr) which we obtain from Tanzil corpora[1]. Furthermore, the language pair German (De) - English (En) is from WMT14[2] and the language pair Chinese (Zh) - En is from LDC[3] corpora. The original training set contains 1.25M but we only use the first 1M sentence pairs as our training set. Meanwhile, we use NIST03 as our validation set and NIST02 as the test set, while the last language pair Vietnamese (Vi) - En is from IWSLT15[4]. All the training corpora are publicly available, and the specifications of the corpora are listed in Table 1. We set the target side to English, and the source sides include morphological rich LRLs. Moreover, for the pair En - De, we combine *newstest2012* and *newstest2013* as the development set, and take *newstest2014* as the test set. Also, for the task of De - En, we split the original data into two different corpus sizes “160K” and “7K” and allocate them for the training set and the validation set separately. Meanwhile, we merge together *dev2020, dev2010, tst2010, tst2011, tst2012* as the test set. Additionally, for the task of Vi -En, we take *tst2012* and *tst2013* as the validation set and the test set, respectively. The process of building vocabulary differs among all language

[1] http://opus.nlpl.eu/Tanzil.php

[2] https://nlp.stanford.edu/projects/nmt/data

[3] The training set includes LDC2002E18, LDC2003E07, LDC2003E14, part of LDC2004T07, LDC2004T08 and LDC2005T06.

[4] https://wit3.fbk.eu

pairs. Precisely, we build the source and target vocabulary using the 32K BPE [5] method [Sennrich et al. 2016b] without jointly training for both Tanzil and NIST corpora and jointly training for both WMT14 and IWSLT15 corpora. Moreover, we build the source and target vocabulary for IWSLT14 by 10K BPE. To train the baseline Bt and the word2vector model, we exploit the monolingual corpora in En, which are provided from WMT17[6]. The original corpus size of WMT17 is 5.9M, but we only use the small partition (1M) which is the same as the corpus size of the language pair Zh - En. We also calculate the statistics of the POS-Tagging labels in our corpora which are given in Table 2. Furthermore, we provide some samples of generated data after augmentation by substitution of various POS-Tagging labels. The original real target sentence is augmented by leveraging the replacement of the name, verb, adjective, and adverb from the target side of Zh - En.

Furthermore, we use the pre-processing *script* to clean up the data by removing duplicated sentences, removing blank lines in the corpus, and cleaning any mismatched sentence lengths. We also use an open-source Chinese word stemmer system THULAC for the Chinese language [Li and Sun 2009]. We leverage the tokenizer toolkit [Koehn et al. 2003] [7] which is provided by the SOTA phrase-based SMT system Moses for word tokenization. We do not use any UNK-replacement techniques [Luong et al. 2015] for the results of any of the experiments. Likewise, in this experiment, we employ the NLTK toolkit [Bird 2006] for tagging the target side monolingual data English in all the corpora. Meanwhile, to obtain the word embeddings [Mikolov et al. 2013] of English and calculate the cosine similarity between tagged words and the words in the paraphrase table, we leverage the gensim toolkit[8]. The paraphrase table[9] contains various corpus sizes of monolingual data with different granularities such as word level, phrase level, and sentence level, but we only use the sizes "S, X, XL" with a word-level corpus of En. In addition, we employ an open-source toolkit FairSeq[10] for the NMT system, together with Transformer architecture (base model), to train and evaluate the baselines Trans and Bt. Meanwhile, for other baselines Swap, Drop, Switch, and SCA, we also take advantage of the FairSeq toolkit. Among the baselines, Switch[11] and SCA[12] use their own codes implemented with FairSeq. Likewise, we run all the experiments for 50 epochs with FairSeq and 100K iter steps with THUMT on 4 GPUs (TITAN X) using default parameters while we slightly revise some of the dimensions (see Table 3). We use the case-sensitive BLEU [13] [Papineni et al. 2002], METEOR[14] score [Banerjee and Lavie 2005] and TER[15] score [Snover et al. 2006] to evaluate the translation performance. We use the pre-trained language model GPT-2 [Radford et al. 2019] to calculate the language perplexity. Moreover, to investigate the quality of augmented data which are generated from different models, we also exploit the ROUGE[16] score [Lin 2004] to further evaluate and compare the quality of our results to other baseline systems.

Baselines. It is best to compare our proposed method with highly similar approaches. Therefore, we select analogous approaches as follows:

[5] https://github.com/rsennrich/subword-nmt
[6] http://www.statmt.org/wmt17/
[7] https://github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer/tokenizer.perl
[8] https://radimrehurek.com/gensim/models
[9] http://nlpgrid.seas.upenn.edu/PPDB/eng/
[10] https://github.com/pytorch/fairseq
[11] https://github.com/cindyxinyiwang/fairseq
[12] https://github.com/teslacool/SCA
[13] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl
[14] https://github.com/meteor/meteor
[15] https://github.com/jhclark/tercom
[16] https://github.com/pltrdy/rouge

Table 3. Hyper-parameter settings.

Parameter	Value
Iter Time	200K
Word Embedding	620
Hidden State	512
Vocabulary Size	30K
Batch Size	80
Sequence Length	50
Beam Size	4
Dropout	0.1
Learning Rate	1.0

In the majority of hyper parameters of different models, we use the default values.

- Trans (Transformer) is the SOTA architecture in NMT system [Vaswani et al. 2017]. It is one of the commonly used and well known architectures in the NMT scenario. In the building of both the HRLs and LRLs systems, the majority of tech engineers and researchers prefer it to previous conventional neural architectures.
- Bt (Back Translation) uses the monolingual data to augment the parallel corpus in MT [Sennrich et al. 2016a]. This training method needs the pre-trained MT model to leverage massive monolingual data to generate the pseudo corpus, and finally combines them with the original corpus to expand the initial size of the corpus. However, it is hard to guarantee that it can obtain higher quality data.
- Swap (Swapping) replaces the words in neighboring positions within a window size k randomly [Artetxe et al. 2018; Lample et al. 2018]. Clearly, the core idea is building the fake data by taking advantage of randomly swapping the words in a sentence. The model consists of a slightly modified attentional encoder-decoder model which can be trained on monolingual corpora alone using a combination of denoising and Bt.
- Drop (Dropping) drops the word tokens arbitrarily from the sequence [Iyyer et al. 2015]. The key idea is randomly dropping the word among the current sentence to build the pseudo corpus. It may make the perplexity higher and higher after augmentation. Namely, they provide the deep averaging network, and it can be improved by applying a novel dropout-inspired regularizer: for each training instance, randomly drop some of the tokens' embeddings before computing the average.
- Switch (Switch-out) samples the data by leveraging the hamming distance sampling method in MT [Wang et al. 2018a]. Precisely, they formulate the design of a data augmentation policy with desirable properties as an optimization problem, and derive a generic analytic solution. They also randomly replace words in both the source sentence and the target sentence with other random words from their corresponding vocabularies.
- SCA (Soft Context Augmentation) data augmentation for NMT replaces a randomly chosen word with a soft distributional representation [Gao et al. 2019] to leverage the bi-lingual language model. More accurately, they replace the one-hot representation of a word by a distribution (provided by a language model) over the vocabulary, i.e., replacing the embedding of this word by a weighted combination of multiple semantically similar words.

Table 4. The comparison with the BLEU score between pipelines on the different partitions of Zh - En.

Method	Zh - En (BLEU) ↑				
	10K	50K	100K	500K	1M
Trans (Vaswani et al. [2017])	4.73	15.73	23.01	29.54	30.61
Bt (Sennrich et al. [2016a])	4.75	15.88	22.95	29.63	30.56
Swap (Artetxe et al. [2018])	4.80	16.01	22.94	29.43	30.59
Drop (Iyyer et al. [2015])	4.84	16.41	23.10	29.75	31.22
Switch (Wang et al. [2018a])	4.89	16.49	23.18	30.23	31.24
SCA (Gao et al. [2019])	**4.94**	**16.53**	**23.28**	**30.40**	**31.34**
Our work					
POS_{NN}	5.12	16.76	23.36	30.51	31.37
POS_{VB}	5.39	16.85	23.43	30.60	31.42
POS_{JJ}	**5.46***	16.88	**23.74***	**30.66***	31.46
POS_{RB}	5.07	**16.99***	23.58	30.63	**31.51***

"$\star$" denotes significantly better than the best baselines with $p < 0.05$.

4.2 Comparison with BLEU score on LDC

As shown in Table 4, we compare our method with the previous six baseline systems on the various sizes of corpus Zh - En from the LDC corpora. To investigate the effectiveness of the proposed method on extremely low-resource languages, we split the original Zh - En into five different corpus sizes of small pieces. Baseline Bt does not achieve better performance continually and it is not enhanced on all the parts of Zh - En. Bt obtains better results than Trans in "10K, 50K and 500K", but the other two pieces are not as good as we expected. Moreover, the recent strong baseline Switch and SCA gain a consistently better result on all the small corpus. Generally, the proposed method achieves better results than all the previous methods on all the various corpus. Especially, replacement of POS_{JJ} ("adjectives") and POS_{RB} ("adverbs") improves the performance of the model frequently.

4.3 Comparison with METEOR Score on LDC

Generally, we also employ another commonly used evaluation matrix *meteor* score for comparing the performance of various models. To further prove the translation quality and robustness of our current best model with other evaluation matrices, we provide the comparison by exploiting the METEOR score over the test set as given in Table 5. It is quite apparent that the strong baseline SCA is extremely distinguished from other pipeline systems. Apart from the results on the 100K, our model achieves similar results with baseline systems on the other partition of the language pair Zh - En. However, our proposed method also significantly outperforms the previous strong baseline with $p < 0.05$. In comparison with other pipeline systems, the proposed model shows improvement in the various divisions of the LDC corpora.

4.4 Comparison with TER Score on LDC

We have mentioned many augmentation tricks for NMT in Session 4.1. To investigate the influence of these methods for the quality of augmented data, we compare their performance using the Translation Error Rate (TER) score, which is another well-known evaluation standard in the MT scenario. As table 6 shows, the commonly used word transformation methods with random sampling, such as shuffling (Swap), dropping, and Bt hurt the model quality. Most of the previous

Table 5. The comparison with METEOR score between baselines for the various partition of Zh - En.

Method	Zh - En (METEOR) ↑				
	10K	50K	100K	500K	1M
Trans (Vaswani et al. [2017])	10.91	21.93	28.83	32.97	34.04
Bt (Sennrich et al. [2016a])	10.95	22.09	28.71	33.00	33.58
Swap (Artetxe et al. [2018])	11.24	22.13	28.88	32.87	33.66
Drop (Iyyer et al. [2015])	11.28	23.41	28.90	33.22	34.05
Switch (Wang et al. [2018a])	11.35	23.65	29.03	33.49	34.24
SCA (Gao et al. [2019])	**11.37**	**23.73**	**29.04**	**33.57**	**34.32**
Our work					
POS$_{NN}$	11.41	23.77	29.05	33.58	34.33
POS$_{VB}$	11.81	23.78	29.06	33.61	34.34
POS$_{JJ}$	11.88	23.88	29.08	**33.92***	34.36
POS$_{RB}$	**11.40**	**23.91**	**29.07**	33.62	**34.38**

Usually, the performance is better when the METEOR score is greater. "★" denotes significantly better than the previous best baselines with $p < 0.05$.

Table 6. The comparison with TER score between different methods for the various partition of Zh - En.

Method	Zh - En (TER) ↓				
	10K	50K	100K	500K	1M
Trans (Vaswani et al. [2017])	92.98	72.82	61.74	54.54	53.04
Bt (Sennrich et al. [2016a])	92.51	72.49	62.76	54.46	53.15
Swap (Artetxe et al. [2018])	92.03	71.34	62.95	55.01	53.07
Drop (Iyyer et al. [2015])	91.33	71.12	61.62	54.43	52.70
Switch (Wang et al. [2018a])	91.18	70.62	61.42	53.70	52.69
SCA (Gao et al. [2019])	**90.77**	**70.57**	**61.25**	**53.51**	**52.54**
Our work					
POS$_{NN}$	90.21	70.10	60.57	53.34	52.51
POS$_{VB}$	87.91	69.61	60.04	53.23	52.46
POS$_{JJ}$	**87.33**$^{++}$	68.89	**59.87**$^{++}$	**52.77***	52.44
POS$_{RB}$	87.34	**68.78**$^{++}$	59.95	52.78	**52.43**

Usually, the performance is better when the TER score is smaller. "++" and "★" denote significantly better the than previous best results with $p < 0.01$ and $p < 0.05$.

tricks ignore the evaluation of the augmented data, or select the sequence just using LM. Our augmentation method is more effective than others. Additionally, it is easy to observe from Table 6 that the stronger baseline Switch and SCA obtain more remarkable results than previous baselines, but are not steady on a different partition of Zh-En. In contrast, our model achieves a consistently better result than all the pipelines on the whole partition of Zh - En from LDC corpora.

4.5 Comparison on Tanzil and IWSLT Corpora

To validate the performance of our approach on other language pairs, we also further validate the proposed approach both on Tanzil and IWSLT (IWSLT14 and IWSLT15) corpora. As given in Table

Table 7. The comparison with the BLEU score between baseline systems on both the Tanzil and IWSLT.

Method	Tanzil				IWSLT14	IWSLT15
	Az - En	Hi - En	Tr - En	Uz - En	De - En	Vi - En
Trans (Vaswani et al. [2017])	21.03	20.15	22.72	17.76	33.53	25.32
Bt (Sennrich et al. [2016a])	21.32	19.20	24.95	18.72	33.69	26.34
Swap (Artetxe et al. [2018])	20.32	21.33	25.08	19.21	33.98	26.98
Drop (Iyyer et al. [2015])	21.19	20.78	25.77	19.30	34.68	27.35
Switch (Wang et al. [2018a])	**26.36**	**22.61**	25.54	19.65	34.75	28.58
SCA (Gao et al. [2019])	25.32	22.16	**25.92**	**19.77**	**34.89**	**29.23**
Our work						
POS$_{NN}$	26.45	22.87	25.98	19.82	34.96	29.65
POS$_{VB}$	26.58	23.01	26.07	20.32	34.91	29.80
POS$_{JJ}$	26.71	24.29	26.00	20.41	34.92	**30.05***
POS$_{RB}$	**26.73***	**24.82***	**26.16***	**20.62***	**35.02**	29.93

"$\star$" denotes significantly better than the previous best baselines with $p < 0.05$.

7, our method obtains better results than other pipeline systems on both the Tanzil and IWSLT corpora. Bt and Swap achieve lower improvements than Trans, but Drop frequently obtained better results on Tanzil and IWSLT. Meanwhile, the Switch and SCA also gain comparable results on these corpora. The SCA obtains better improvements on all the corpus except "Az - En" and "Hi - En". In contrast, our proposed approach continually achieves significantly better performance than all the baselines. It is also easy to infer from the experiment results that the replacement of "adverbs" almost obtains better results than "adjective" on De - En.

4.6 Ablation Study

Our model consists of two parts: POS-Tagging replacement and selection of the various lengths of paraphrase table. Generally, it is clear to infer that these two main factors may affect the generalization skill of our proposed method. Therefore, here we make some investigations on these factors.

Effect of Different POS-Tagging Replacement. As shown in Table 8, to select and examine the most reasonable replacement method for this work, we further investigate the effect of various replacing approaches between all the corpora. In this experiment, we only substitute a word with a POS-Tags once (see Session 3) instead of replacing the POS-Tags a couple of times in the same sentence. For instance, if there are two or more nouns, we only substitute one word among them once at a time rather than replacing all the nouns at once. Moreover, in the POS$_{NN-VB}$ experiment, we do it in three steps: First, we replace a noun in the current sentence to generate a new corpus. Then, we replace a verb in the same sentence to build another new corpus. Finally, we combine the two generated corpus to build the final bigger new corpus. The replacing method POS$_{Tag1-Tag2-Tag3}$ (replace three tags) achieves remarkable results on both Tanzil and LDC (Zh - En$_{100K}$) corpora, but in the IWSLT it did not achieve better results. The replacing method POS$_{Tag1-Tag2}$ (replace two tags) obtains comparable results than other methods on IWSLT. Additionally, it implies from the experiment results that, as long as the replacement combination contains the "adverb and adjective", it helps the model enhance performance.

Table 8. The effect of different POS-Tagging replacement on the Tanzil, IWSLT, and LDC corpora with BLEU.

Method	Tanzil			IWSLT14	IWSLT15	LDC
	Az - En	Hi - En	Uz - En	De - En	Vi - En	Zh - En$_{100K}$
POS$_{NN}$	26.45	22.87	19.82	34.96	29.65	23.36
POS$_{VB}$	26.58	23.01	20.32	34.91	29.80	23.43
POS$_{JJ}$	26.71	24.29	20.41	34.92	**30.05**	**23.74**
POS$_{RB}$	**26.73**	**24.82**	**20.62**	**35.02**	29.93	23.58
POS$_{NN-VB}$	26.02	25.50	21.23	34.55	29.96	24.06
POS$_{NN-JJ}$	26.58	25.08	20.73	34.84	30.03	23.82
POS$_{NN-RB}$	26.87	24.49	20.71	35.04	29.84	24.11
POS$_{VB-JJ}$	25.56	24.76	20.32	35.16	**30.15**	24.01
POS$_{VB-RB}$	26.76	25.23	21.47	34.98	30.10	23.89
POS$_{JJ-RB}$	26.88	25.56	20.88	**35.39***	29.92	23.93
POS$_{NN-VB-JJ}$	26.90	24.43	21.55	35.35	29.77	23.85
POS$_{NN-VB-RB}$	26.91	**26.99**$^{++}$	20.98	34.75	30.13	**24.19**$^{++}$
POS$_{NN-JJ-RB}$	26.93	24.82	20.91	34.68	30.09	23.98
POS$_{VB-JJ-RB}$	**27.09***	25.11	**21.79**$^{++}$	35.32	28.90	23.90

The six rows at the middle and the last four rows stand for replacing the two and four different POS-Tagging labels at the same time. Moreover, " Zh - En$_{100K}$" stands for the medium partition of the Zh - En corpus whose corpus size is 100K. "++" and "★" denote significantly better than previous best results with $p < 0.01$ and $p < 0.05$.

Table 9. The investigation of the same POS-Tagging replacement with various corpus size of paraphrase table.

Method	Tanzil (BLEU [%])		
	Az - En	Hi - En	Uz - En
Paraphrase$_{T-S}$	**25.89**	**25.56**	**20.42**
Paraphrase$_{T-X}$	26.38	26.23	21.10
Paraphrase$_{T-XL}$	**27.09**$^{++}$	**26.99**$^{++}$	**21.79**$^{++}$

The subscripts "T-S, T-X, T-XL" represent the different corpus sizes. They are small, large, and larger, respectively. "++" denotes significantly better than previous best results with $p < 0.01$.

Effect of Various Length of Paraphrase Table. As shown in Table 9, we also further investigate the effectiveness of the different sizes of the paraphrase table. We find that a bigger paraphrase table usually provides a larger embedding space for searching. Meanwhile, the bigger corpus size of the paraphrase table makes the augmentation model stronger than before. As given in Table 8, the performance of POS$_{VB-JJ-RB}$ achieves better improvements than other combinations of replacement on Tanzil corpora. Therefore, to further examine the effectiveness of various corpus sizes of paraphrase table, we choose the same tag (POS$_{VB-JJ-RB}$).

Effect of Random Replacement with Paraphrase Table. As shown in Table 10, we also investigate the effect of random paraphrasing on the translation quality of our proposed appraoch. We explore the different replacing strategies by using POS-Taggings and random substitutions based on paraphrase

Table 10. The effect of random paraphrasing replacement on the Tanzil, IWSLT, and LDC corpora with BLEU.

Method	Tanzil			IWSLT14	IWSLT15	LDC
	Az - En	Hi - En	Uz - En	De - En	Vi - En	Zh - En$_{100K}$
Baseline	26.36	22.61	19.77	34.89	29.23	23.28
Paraphrase + Random	26.01	25.16	20.49	34.90	29.24	23.33
Paraphrase + POS-Taggings	**27.09**$^{++}$	**26.99**$^{++}$	**21.79**$^{++}$	**35.39***	**30.15**$^{++}$	**24.19**$^{++}$

"Baseline" represents the best results of the strong baselines from Table 4 and 7. "Paraphrase + POS-Taggings" denotes replacing the words with respect to the POS-Taggings via paraphrase table, and "Paraphrase + Random" represents substitute the randomly selected words using paraphrase table. "++" and "$\star$" denote significantly better than previous best results with $p < 0.01$ and $p < 0.05$.

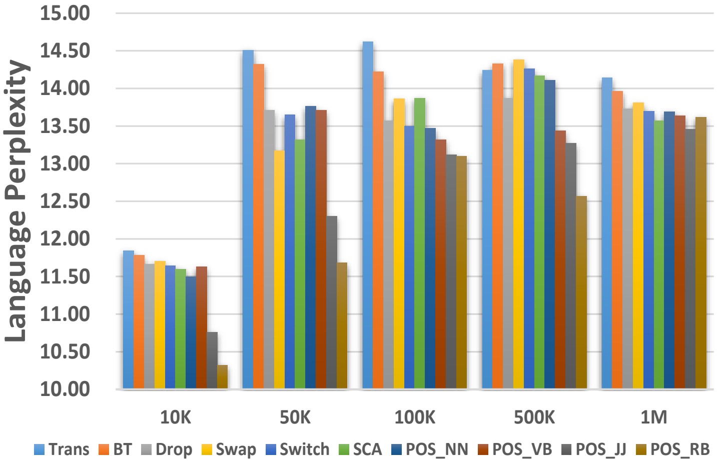


Fig. 4. The comparison with language perplexity (the smaller the better) for augmenting the target side of the language pair Zh - En from LDC corpus.

table. The results from random paraphrasing are better than the strong baselines but lower than the POS-Tagging-based method. We achieve the higher improvements from POS-Tagging-based replacement compared to randomly paraphrasing and strong baseline system. Our POS-Tagging-based replacement consistently obtained the performance than the baseline and random paraphrasing manner on all language pairs except De - En. Therefore, we take the POS-Taggings-based method as our final augmentation approach instead of using the random paraphrasing method.

4.7 Comparison with Perplexities (PPL)

To validate the adequacy and the fluency of the translation result, and to further consider the effectiveness of the proposed model we compared our results to other baseline systems by using the language perplexity. As illustrated in Figure 4, our method achieve better performance than all the baseline systems on LDC corpora. Contrarily, Drop achieves better improvements under all settings except 50K, and Switch and SCA also gain better results in all small corpus. Obviously, no previous methods consistently obtain better results, and our method outperforms other baselines on all low-resource language pairs. The performances of POS$_{RB}$ and POS$_{JJ}$ are quite remarkable.

Table 11. The comparison with ROUGE score [$Rouge - 1$] on Zh - En$_{100K}$.

Method	ROUGE score ↑ (Rouge - 1)		
	Precision	Recall	F-measure
Trans (Vaswani et al. [2017])	57.18	57.25	56.75
Bt (Sennrich et al. [2016a])	57.11	57.28	56.72
Swap (Artetxe et al. [2018])	57.13	56.22	56.24
Drop (Iyyer et al. [2015])	57.21	56.22	56.29
Switch (Wang et al. [2018a])	57.32	56.28	56.37
SCA (Gao et al. [2019])	**57.98**	**55.43**	**56.21**
Our work			
POS$_{NN}$	58.31	55.49	56.41
POS$_{VB}$	58.39	55.49	56.35
POS$_{JJ}$	**58.41***	**55.72***	**56.62***
POS$_{RB}$	58.40	55.62	56.56

Usually, the performance is better when the ROUGE score is greater. "$\star$" denotes significantly better than previous best results with $p < 0.05$.

Table 12. The comparison with ROUGE score [$Rouge - 2$] on Zh - En$_{100K}$.

Method	ROUGE score ↑ (Rouge - 2)		
	Precision	Recall	F-measure
Trans (Vaswani et al. [2017])	29.94	29.31	29.40
Bt (Sennrich et al. [2016a])	29.73	29.11	29.19
Swap (Artetxe et al. [2018])	29.74	29.14	29.21
Drop (Iyyer et al. [2015])	**30.50**	29.06	29.48
Switch (Wang et al. [2018a])	29.84	29.87	29.60
SCA (Gao et al. [2019])	29.95	**29.93**	**29.68**
Our work			
POS$_{NN}$	30.60	29.19	29.62
POS$_{VB}$	30.97	29.40	30.00
POS$_{JJ}$	**31.31***	**29.80**	**30.30***
POS$_{RB}$	31.21	29.63	30.17

Usually, the performance is better when the ROUGE score is greater. "$\star$" denotes significantly better than previous best results with $p < 0.05$.

4.8 Comparison with ROUGE Scores on LDC

As given in Table 11, to further validate the effectiveness of adequacy and fluency, we also employ another evaluation matrix of rouge to evaluate the quality of augmented data. This evaluation matrix is widely used in dialogue generation tasks. DA also is one of the text generation tasks, therefore we exploit this matrix to evaluate the adequacy and fluency of augmented data. According to $Rouge - 1$ score, we can infer that our proposed method significantly outperforms the previous six baseline systems. As shown in Table 12, we also further compare the effectiveness of these approaches for the performances of generated pseudo data using the $Rouge - 2$ score on the language pair of Zh - En, and our model obtain more remarkable results than others. As illustrated in the line chart in

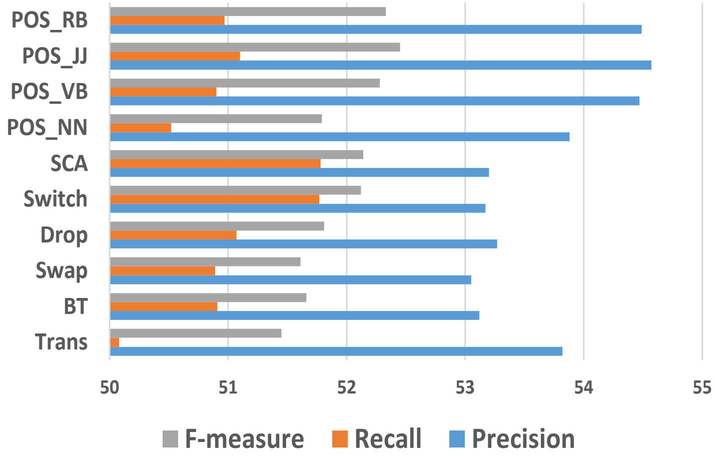


Fig. 5. The comparison with ROUGE score [$Rouge - L$] (the greater the better) for augmenting the target side of the language pair Zh - En$_{100K}$ from LDC corpus.

Figure 5, the stronger baselines Switch and SCA also obtained more effective results than previous baseline systems, but the model gains more effective results with "Precision" and "F-measure" than others on Zh - En with $Rouge - L$ score. Our replacement method obtains remarkably better enhancements than other baselines.

4.9 Case Study

As shown in Table 13, the previous methods Trans and Bt translate some words ("set up" and "recruited") in the source sequence, but the translation results are still incorrect. Moreover, the method Swap also translates better than both Trans and Bt, but incorrectly translates the phrase ("a group of") into "a group". They also cause syntactic and semantic problems. Meanwhile, the Drop method also adds extra words and makes some mistakes in the translation of tenses. Switch and SCA also translate incorrectly on the tense problem. In comparison to previous baseline systems, our approach reduces both syntactic and semantic errors to some extent. Meanwhile, the proposed method can deal with the translation of tense problems and generate accurate results compared to the reference sequence.

5 RELATED WORK

As a key potential strategy for overcoming the grave problem posed by the shortage of large-scale parallel corpora, NMT has gained increasing attention in the MT community [Dong et al. 2015; Firat et al. 2016a,b; Sennrich et al. 2016b; Vaswani et al. 2017]. Many approaches have been proposed that aim to address the issue of deficiency of the parallel training corpora in NMT for LRLs. Most of the current literature can be classified into meta-learning [Gu et al. 2018], transfer learning [Mieradilijiang and Xiaohui 2018; Tan et al. 2017; Zoph et al. 2017, 2016], domain adaptation [Chu et al. 2017], pivot learning [Cheng et al. 2017], "teacher-student network" [Chen et al. 2017], "semi-supervised" [Cheng et al. 2016] and zero-shot learning [Chen et al. 2017]. Additionally, there are also a variety of techniques for data augmentation in the NMT scenario. To our knowledge, these approaches are categorized into three types. The first type is based on BT which leverages the monolingual data to augment the training corpus guided by the trained MT model. It is effective, but usually helpless to errors. The second type is word transformation with or without a language model,

Table 13. Translation examples of the test set from Zh - En$_{1M}$ between different DA methods.

Method	Translation result
Source	*qunian , yi daxing qiye zai zhu sanjiao mou di sheli fengongsi , zhao le yi pi daxuesheng .* 去年 , 一 大型 企业 在 珠 三角 某 地 设立 分公司 , 招 了 一 批 大学生 。
Reference	last year , a large **corporation** set up a branch in the pearl river delta and recruited **a group of** college students .
Trans (2017)	*last year , bigger company **set** the branch in the pearl river delta, recruits group of **university staff** .*
Bt (2016a)	*last year , large company sets the **subset** in the pearl delta and recruit the group of college students .*
Swap (2018)	*last year , **the** large company **find** a branch pearl river delta and recruited **a group** college students .*
Drop (2015)	*last year , big corporation **make** a branch in the pearl river delta and **recruit** a bunch of **university** student .*
Switch (2018a)	***past year** , a big corporation **making** branch in pearl river delta, **recruit** number of college student **recruit** .*
SCA (2019)	*last year , a large corporation **create** a part in the pearl delta and recruited a countless of **university** students .*
Our work	
POS$_{NN}$	*The **past** year , large company **made** a branch in the pearl river delta, recruit **bunch** of college students .*
POS$_{VB}$	*last year , **the** large corporation construct branch in pearl river delta , employed **number** of college students .*
POS$_{JJ}$	*last year , a big corporation **created** the branch in pearl river delta , engaged **many** college students .*
POS$_{RB}$	*last year , the large company found a branch in the pearl river delta , recruited **a numbers of** college students .*

“ Zh - En$_{1M}$" represents the biggest partition of the Zh - En corpus whose corpus size is 1M.

e.g., it uses swapping [Artetxe et al. 2018], replacing [Fadaee et al. 2017], switching [Kobayashi 2017], omitting [Lample et al. 2018] and context knowledge [Gao et al. 2019] methods. The third type is based on various sampling methods. A recent work [Wang et al. 2018a] follow the idea of Norouzi et al. [2016] and propose an effective method which uses hamming distance sampling, and they take the DA as an optimization problem.

Intuitively, the NMT systems have been developed rapidly in recent years. Zeng et al. [2018] enhance the performance of the NMT using word-level domain context in the multi-domain community. Meanwhile, Zeng et al. [2019] present the iterative dual training method for domain adaptation task in NMT. Moreover, Su et al. [2019] commit to distinguishing, and exploiting different word-level domain contexts for multi-domain NMT, and improve the NMT model generalization skill by exploiting multi-task learning to jointly model NMT and monolingual attention-based domain classification tasks. In the past two years, many researchers have proposed new approaches [Fadaee et al. 2017] in the NMT scenario for LRLs. Maimaiti et al. [2019] propose the multi-round transfer learning for LRLs in NMT. Gu et al. [2018] present the meta-learning for low-resource

NMT and achieve a remarkably better result. Sennrich and Zhang [2019] discuss some pitfalls to be aware of when training the low-resource NMT systems, and recent techniques that have shown to be especially helpful in low-resource settings.

Also, Ke et al. [2019] take advantage of the edit distance sampling method for dialogue generation. Wei and Zou [2019] use synonym replacement, random insertion, random swapping and random deletion to achieve boosting performance on text classification. A paraphrase augmented response generation framework [Gao et al. 2020] improves the dialog generation performance by exploiting paraphrase. Furthermore, some methods with a combination of BT [Sugiyama and Yoshinaga 2019; Sun et al. 2020] have been proposed. The tricks belong to the previous two types, more or less, and they ignore the reduction of errors during or after augmentation. The methods in the last type (e.g. hamming-distance sampling) replace the words based on uniform distribution. In contrast, we consider target side augmentation. Besides we do not need to evaluate the network to filter high quality augmented data after generation. Moreover, our augmentation method samples better sentences before transmitted to the evaluation model since we use POS-Tagging and paraphrase word embeddings to replace words which are selected by cosine similarity and ranking rather than random selection.

6 CONCLUSION AND FUTURE PERSPECTIVE

In this work, we propose quite a straightforward and effective DA approach for LRLs in NMT, which augments the pseudo sequence from the original real sample by using POS-Tagging and the paraphrase embedding. Besides, we investigate both the effect of different combinations of POS-Tagging and the corpus size of the paraphrase table. The experimental results including PPL and rouge score show that our method significantly outperforms all the previous methods. Moreover, since we exploit the POS-Tagging labels they help our model generate a highly analogous sequence with even lower syntactic errors. Besides, the cosine similarity during the ranking step also mitigates the semantic error when the selected word is being replaced.

Therefore, we do not design a special evaluator for augmented data. As shown in all the experiment results, the proposed method is rather straightforward but effective for NMT in LRLs. Likewise, it is easy to infer from the result of language perplexities and rouge evaluation matrices, our model has better adequacy and fluency than previous approaches. It also demonstrates that our model is good enough to reduce errors during augmentation. Our method is model transparent and language-independent. Therefore we can incorporate it as a replacement method into various MT architectures in several languages.

In the future, we will use this method for other text generation tasks, such as information retrieval, speech recognition, summarization, dialogue generation, chat-bot, question answering, and Knowledge Graphs apart from the MT tasks. Meanwhile, we will refine our approach using bilingual word embeddings [Su et al. 2018]. Besides, we would like to examine the effect of other factors on augmentation, as well as the effectiveness of various combinations of POS-Tagging, and we will also investigate the influences of even bigger corpus size on the paraphrase table.

ACKNOWLEDGMENTS

We would like to thank all anonymous reviewers for their valuable comments and suggestions during the minor revision for this work. This work was supported by the National Key R&D Program of China (No. 2017YFB0202204), National Natural Science Foundation of China (No. 61925601, No. 61772302) and Sogou Inc. Besides, the authors also would like to thank Dr. DO HA LOC who is the assistant researcher at the multi-lingual team of DAMO academy in Alibaba Group, for patiently proofreading our paper before submission. We also would like to thank Mike Madden, MBA candidate at Johns Hopkins for proofreading after the minor revision was complete.

REFERENCES

M. Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised Neural Machine Translation. In *Proceedings of ICLR*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. In *NIPS*.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of ICLR*.

S. Banerjee and A. Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *IEEvaluation@ACL*.

Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *COLING/ACL*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* (1993).

Yun Chen, Yang Liu, Yong Cheng, and Victor O.K. Li. 2017. A Teacher-Student Framework for Zero-Resource Neural Machine Translation. In *Proceedings of ACL*.

Yun Chen, Yang Liu, and Victor O. K. Li. 2018. Zero-Resource Neural Machine Translation with Multi-Agent Communication Game. *CoRR* abs/1802.03116 (2018).

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-Superivsed Learning for Neural Machine Transaltion. In *Proceedings of ACL*.

Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. 2017. Joint Training for Pivot-based Neural Machine Translation. In *Proceedings of IJCAI*.

David Chiang. 2005. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of ACL*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of EMNLP*.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An Empirical Comparison of Simple Domain Adaptation Methods for Neural Machine Translation. *CoRR* abs/1701.03214 (2017).

E. Cubuk, Barret Zoph, Dandelion Mané, V. Vasudevan, and Quoc V. Le. 2019. AutoAugment: Learning Augmentation Strategies From Data. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 113–123.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task Learning for Multiple Language Translation. In *Proceedings of ACL*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data Augmentation for Low-Resource Neural Machine Translation. In *ACL*.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. In *Proceedings of NAACL*.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016b. Zero-Resource Translation with Multi-Lingual Neural Machine Translation. In *Proceedings of EMNLP*.

Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5539–5544.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase Augmented Task-Oriented Dialog Generation. In *Proceedings of ACL*.

Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and V. Li. 2018. Meta-Learning for Low-Resource Neural Machine Translation. In *EMNLP*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the CVPR*. 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* (1997).

Mohit Iyyer, V. Manjunatha, Jordan L. Boyd-Graber, and Hal Daumé. 2015. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In *ACL*.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions. arXiv:1610.01108v2. (2016).

Pei Ke, Fei Huang, Minlie Huang, and Xiaoyan Zhu. 2019. ARAML: A Stable Adversarial Training Framework for Text Generation. In *EMNLP/IJCNLP*.

S. Kobayashi. 2017. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In *NAACL*.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical Phrase-based Translation. In *Proceedings of NAACL*.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised Machine Translation Using Monolingual Corpora Only. In *ICLR*.

Zhongguo Li and Maosong Sun. 2009. Punctuation as Implicit Annotations for Chinese Word Segmentation. *Computational Linguistics* 35 (2009), 505–512.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://www.aclweb.org/anthology/W04-1013

Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the Rare Word Problem in Neural Machine Translation. In *ACL*.

M. Maimaiti, Y. Liu, Huanbo Luan, and M. Sun. 2019. Multi-Round Transfer Learning for Low-Resource NMT Using Multiple High-Resource Languages. *ACM Trans. Asian Low Resour. Lang. Inf. Process.* 18 (2019), 38:1–38:26.

Maimaiti Mieradilijiang and Zou Xiaohui. 2018. Discussion on Bilingual Cognition in International Exchange Activities. In *ICIS2018*.

Tomas Mikolov, Kai Chen, G. S. Corrado, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013).

Mohammad Norouzi, S. Bengio, Z. Chen, Navdeep Jaitly, Mike Schuster, Y. Wu, and Dale Schuurmans. 2016. Reward Augmented Maximum Likelihood for Neural Structured Prediction. In *NIPS*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a methof for automatic evaluation of machine translation. In *Proceedings of ACL*.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Rico Sennrich, B. Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of ACL*.

Rico Sennrich and Biao Zhang. 2019. Revisiting Low-Resource Neural Machine Translation: A Case Study. In *ACL*.

Matthew Snover, Bonnie J. Dorr, R. Schwartz, and L. Micciulla. 2006. A Study of Translation Edit Rate with Targeted Human Annotation.

Jinsong Su, Shan Wu, Biao Zhang, Changxing Wu, Yue Qin, and Deyi Xiong. 2018. A neural generative autoencoder for bilingual word embeddings. *Inf. Sci.* 424 (2018), 287–300.

Jinsong Su, Jiali Zeng, John Xie, H. Wen, Yongjing Yin, and Y. Liu. 2019. Exploring Discriminative Word-Level Domain Contexts for Multi-domain Neural Machine Translation. *IEEE transactions on pattern analysis and machine intelligence* (2019).

Amane Sugiyama and Naoki Yoshinaga. 2019. Data augmentation using back-translation for context-aware neural machine translation. In *DiscoMT@EMNLP*.

Yibo Sun, Duyu Tang, N. Duan, Yeyun Gong, X. Feng, B. Qin, and D. Jiang. 2020. Neural Semantic Parsing in Low-Resource Settings with Back-Translation and Meta-Learning. In *AAAI*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of NIPS*.

Ben Tan, Yu Zhang, Sinno Jialin Pan, and Qiang Yang. 2017. Distant Domain Transfer Learning. In *AAAI*.

Clara Vania and Adam Lopez. 2017. From Characters to Words to in Between: Do We Capture Morphology?. In *ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of NIPS*.

Rui Wang, M. Utiyama, A. Finch, L. Liu, Kehai Chen, and Eiichiro Sumita. 2018b. Sentence Selection and Weighting for Neural Machine Translation Domain Adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26 (2018), 1727–1741.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018a. SwitchOut: an Efficient Data Augmentation Algorithm for Neural Machine Translation. In *EMNLP*.

Yiren Wang, Lijun Wu, Yingce Xia, Tao Qin, ChengXiang Zhai, and T. Liu. 2020. Transductive Ensemble Learning for Neural Machine Translation. In *AAAI*.

Jason Wei and K. Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *EMNLP/IJCNLP*.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and S. Hu. 2019. Conditional BERT Contextual Augmentation. In *ICCS*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR* abs/1609.08144 (2016).

Ziang Xie, Sida I. Wang, J. Li, Daniel Lévy, Allen Nie, Dan Jurafsky, and A. Ng. 2017. Data Noising as Smoothing in Neural Network Language Models. In *ICLR*.

Jiali Zeng, Y. Liu, Jinsong Su, Yubing Ge, Yaojie Lu, Yongjing Yin, and Jiebo Luo. 2019. Iterative Dual Domain Adaptation for Neural Machine Translation. In *EMNLP/IJCNLP*.

Jiali Zeng, Jinsong Su, H. Wen, Yang Liu, J. Xie, Yongjing Yin, and J. Zhao. 2018. Multi-Domain Neural Machine Translation with Word-Level Domain Context Discrimination. In *EMNLP*.

Jiajun Zhang, Y. Zhao, Haoran Li, and C. Zong. 2019. Attention With Sparsity Regularization for Neural Machine Translation and Summarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27 (2019), 507–518.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. 2017. Learning Transferable Architectures for Scalable Image Recognition. *CoRR* abs/1707.07012 (2017).

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of EMNLP*.

Received January 2021; revised March 2021; accepted May 2021