English-to-Japanese Diverse Translation by Combining Forward and Backward Outputs

Masahiro Kaneko Aizhan Imankulova Tosho Hirasawa Mamoru Komachi
Tokyo Metropolitan University
6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan
{kaneko-masahiro, imankulova-aizhan, hirasawa-tosho}@ed.tmu.ac.jp
komachi@tmu.ac.jp

Abstract

We introduce our TMU system that is submitted to The 4th Workshop on Neural Generation and Translation (WNGT2020) to English-to-Japanese (En→Ja) track on Simultaneous Translation And Paraphrase for Language Education (STAPLE) shared task. In most cases machine translation systems generate a single output from the input sentence, however, in order to assist language learners in their journey with better and more diverse feedback, it is helpful to create a machine translation system that is able to produce diverse translations of each input sentence. However, creating such systems would require complex modifications in a model to ensure the diversity of outputs. In this paper, we investigated if it is possible to create such systems in a simple way and whether it can produce desired diverse outputs. In particular, we combined the outputs from forward and backward neural translation models (NMT). Our system achieved third place in En→Ja track, despite adopting only a simple approach.

1 Introduction

WNGT2020[1] on STAPLE[2] (Mayhew et al., 2020) addresses generating high-coverage sets of plausible translations which can be useful in machine translation (MT), MT evaluation, multilingual paraphrase, and language education technology fields. In Duolingo (the world's largest language learning platform), some learning takes place via translation-based exercises and assessment is done by comparing the learners' responses to a large set of acceptable human-generated translations. Therefore, retaining richer paraphrases of the translation results would help to generate more accurate feedback to the learners.

[1] https://sharedtask.duolingo.cites.google.com/view/wngt20/home
[2] https://sharedtask.duolingo.com/

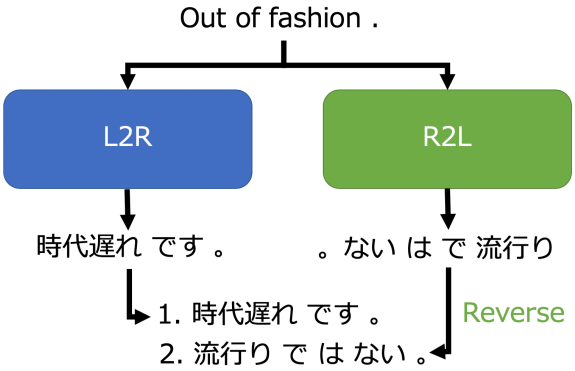

Figure 1: Architecture of TMU system.

Several studies have been conducted on the diversity of translation results (Vijayakumar et al., 2018; Xu et al., 2018; Shu et al., 2019; Ippolito et al., 2019). On the other hand, these methods rely on complex approaches. For example, modifying beam-search (Vijayakumar et al., 2018), introducing rewriting patterns or sentence codes (Xu et al., 2018; Shu et al., 2019) or using post-decoding clustering (Ippolito et al., 2019). However, we were curious if we can produce diverse outputs only using a simple approach.

Therefore, we aim to generate a variety of translations simply using generally adopted neural MT (NMT) methods. For that purpose, we use the models trained on the left-to-right (L2R) and right-to-left (R2L) directions, where L2R produces target sentences in a forward way and R2L will produce target sentences in a backward way as shown in Figure 1. We then combine the output of L2R and back-reversed output of R2L to produce diverse translations. We adopt this approach based on the following reasons:

- No need to modify the NMT model.
- Reversing only the target sentences is sufficient.

Proceedings of the 4th Workshop on Neural Generation and Translation (WNGT 2020), pages 134–138

www.aclweb.org/anthology/D19-56%2d

- It is known that L2R translates prefixes and R2L translates suffixes better (Liu et al., 2016). This indicates that L2R and R2L produce different translation results, which may have an impact on the diversity of generated translations.

In our experiments, we show that even the combination of L2R and R2L translation results can produce a sufficiently diverse set of translations. In addition, we demonstrate that even though we use a simple approach, it is possible to generate varied paraphrased transcriptions which do not simply replace one word with another, contrarily, it utilizes different styles, opposition, word order etc. Our TMU system achieved the third place using only the simple approach.

2 Related Work

Several models have been proposed to generate diverse decoding outputs for different tasks. For example, Xu et al. (2018) proposed diverse paraphrase generation by introducing rewriting patterns into the decoder of the encoder-decoder model. Vijayakumar et al. (2018) proposed diverse beam search algorithm for decoding diverse sequences. They describe beam search as an optimization problem and augment the objective with a diversity term. They encouraged diversity between beams at each step by rewarding each group for spending its beam budget to explore different parts of the output space rather than repeatedly chasing sub-optimal beams from prior groups. They report their results on image captioning, visual question generation, and MT tasks. Shu et al. (2019) generated diverse translations by conditioning sentence generation with the sentence codes. They explored two methods: (a) semantic coding model which extracted sentence codes from unsupervisedly learned semantic information and (b) syntactic coding model which derived the sentence codes from the parse trees produced by a constituency parser. Ippolito et al. (2019) proposed the use of over-sampling followed by post-decoding clustering to remove similar sequences. They evaluated several techniques on an open-ended dialog task and image captioning task.

These works introduce different complex modifications to the model in order to achieve diversity while generating the output. However, in this paper, we show how to simply generate diverse outputs.

Pre-train	
Model Architecture	Transformer-big
Number of epochs	20
Max tokens	4,096
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98$, $\epsilon = 1 \times 10^{-8}$)
Learning rate	5×10^{-4}
Learning rate schedule	inverse sqrt
Warmup updates	4,000
Min learning rate	1×10^{-9}
Loss function	label smoothed cross-entropy ($\epsilon_{ls} = 0.1$) (Szegedy et al., 2016)
Dropout	0.3
Gradient Clipping	0.1
Fine-tuning	
Number of epochs	10
Learning rate	3×10^{-5}
Learning rate schedule	fixed
Translation	
Beam size	64
Ensemble	4

Table 1: Hyperparameter values of NMT model.

3 Experiments

3.1 System

We used the open-source `fairseq`[3] (Ott et al., 2019) for training NMT models. We adopt the Transformer (Vaswani et al., 2017) as our translation model. We train two types of models, L2R and R2L for decoding. For L2R, we train a forward model in a traditional way. For R2L model, we first reverse the target sentences and train a model so it will produce the output from backward. Then the output of R2L is reversed again to forward direction. We exclude sentences from the translation results by normalizing the log probabilities of the hypothesis sentences by sentence length with less than -1.55 score. Then, the n-best translation results of each L2R and R2L are combined and if there is a duplication, one of the translations is removed.

In our preliminary experiments, we found that the NMT model cannot produce sufficient quality translations using only the official data set. Therefore, we pre-train the NMT models with additional datasets, followed by fine-tuning with the STAPLE dataset. Thus, we expect that a model learns

[3] https://github.com/pytorch/fairseq

Data	Size
Official STAPLE-train	2,500 / 855,941
Official STAPLE-dev	500 / 172,817
Official STAPLE-test	500 / 165,095
STAPLE-train	2,450 / 837,879
STAPLE-dev	50 / 18,062
OpenSubtitles	2,083,600
Tatoeba	202,167
TED-train	152,115
TED-dev	1,958
TED-test	1,982

Table 2: Statistics on official STAPLE data and data used in our experiments. For STAPLE data, the left side indicates the number of prompts and the right side indicates the total number of sentences contained in each prompt.

general translation ability during pre-training and further learns to produce more diverse translation during fine-tuning.

3.2 Hyperparameters

Table 1 lists some specific hyperparameters used in our experiments. For fine-tuning, we used the same values as we used for pre-training regarding the values that are not listed in the table. We trained four L2R models and four R2L models with different seeds on the same data, then ensembled all of them by taking the union of their outputs. We adjusted the hyperparameters using the development set, described in the next subsection.

3.3 Data

Table 2 summarizes the size of data used in our experiments for En→Ja track. The official dataset of STAPLE contains multiple translations for a single prompt. We did not use the official development and test data in our experiments because the correct data with answers were not available to the public. Therefore, we randomly divided the official training data into training data and development data in prompt units as shown in Table 2. We use OpenSubtitles[4] (Lison and Tiedemann, 2016), Tatoeba[5] (Tiedemann, 2012), TED[6] train and dev (Cettolo et al., 2012) corpora as additional dataset which are similar to the STAPLE data in terms of sentence length and data domain. We used STAPLE-train, OpenSubtitles, Tatoeba and TED-train as training data and STAPLE-dev, TED-dev and TED-test as development data for the pre-training. In fine-tuning, we used STAPLE-train as training data and STAPLE-dev as development data.

We lowercased all the English data. English was tokenized using *tokenizer.perl* of `Moses`[7] (Koehn et al., 2007) and Japanese was tokenized using `MeCab`[8] with the IPA dictionary. After tokenization, we adopted sub-word segmentation mechanism (Sennrich et al., 2016)[9]. Note that, for the training of R2L, we first applied tokenization for the target sentences, then applied sub-word segmentation and then performed the reversing. The size of the sub-word vocabularies was set to 8,000. The sub-word vocabularies were constructed using pre-train training data.

System	F1
jbrem	31.8
sweagraw	29.4
TMU	**28.3**
mzy	26.0
hzguo	23.9
jindra.helcl	21.3
darkside	19.4
STAPLE_aws_baseline	4.3
STAPLE_fairseq_baseline	3.3

Table 3: The official results on the test set for En→Ja in terms of weighted F1.

Model	F1
Single seed 1	23.7
Single seed 2	23.4
Multi seed	23.9
L2R	23.7
R2L	23.2
L2R & R2L	**24.7**

Table 4: The result for each model in terms of weighted F1 on the development set.

[4] http://opus.nlpl.eu/OpenSubtitles-v2018.php
[5] http://opus.nlpl.eu/Tatoeba.php
[6] https://wit3.fbk.eu
[7] https://github.com/moses-smt/mosesdecoder
[8] http://taku910.github.io/mecab
[9] https://github.com/rsennrich/subword-nmt

Source	your skirt is out of fashion.	
Output 1	あなたのスカートは時代遅れである。	(Your skirt is outdated.)
Output 2	あなたのスカートは流行していない。	(Your skirt is not in fashion.)
Source	they give me water.	
Output 1	彼女らは私に水をくれる。	(They give me water.)
Output 2	私は彼らから水をもらいます。	(I get water from them.)
Source	she found another path.	
Output 1	彼女は違う道を見つけた。	(She found a different path.)
Output 2	彼女は別の道を見つけたわ	(She found another way)
Output 3	彼女は別の道を見つけたよ	(She found another way)

Table 5: Examples generated by the combination of four ensemble L2R and four ensemble R2L models' outputs using the development set. () indicate their English translation. The English translation of the third example can not fully represent the change of styles used in Japanese language output.

3.4 Results

We used weighted macro F1 as the main scoring metric (Mayhew et al., 2020). The system is scored based on its ability to return all acceptable human-made translations, weighted by the likelihood that the learner will respond to each translation. The weighted macro F1 calculates a weighted F1 for each prompt and takes the average of all the prompts in the corpus.

Table 3 lists the F1 scores of participating systems in En→Ja track. Our TMU system was ranked the third.

4 Discussion

4.1 Does translation in opposite directions contribute to a diverse translation?

We investigate whether decoding in opposite directions contribute to diversity in translation outputs. We compare the results for development set generated with beam size of 64 in one model (L2R, R2L, Single seed 1, Single seed 2) to those generated and combined two models (L2R & R2L, Multi seed) with beam size of 32. As a baseline, we also experiment with different seeds and examine their efficiency. This allows us to see how the direction or seed contribute to the diversity of translation.

Table 4 shows the results for top-2 single seeds models in terms of performance and multi seed model, and the best L2R, R2L, and L2R & R2L models. The results show that using multiple seeds leads to higher F1 scores, however, the improvement is not critical. On the other hand, L2R & R2L improved weighted F1 scores for 1.0 points. Therefore, we show that it is important to combine the outputs of the two directions.

4.2 Examples of Translations

Table 5 demonstrates the example of diverse translations generated by the combination of four ensemble L2R and four ensemble R2L models' outputs. Here we sampled the outputs from development set. The first example illustrates how our system uses negation to express the same meaning translations of the source sentence. The second example changed the syntax by using benefactive verbs for the output while preserving the same meaning and grammatical correctness. The third example uses different styles, which are specific for Japanese language, to introduce diversity.

Therefore, we can conclude that even using simple approach we can achieve diverse, grammatically correct translations without changing the meaning of the input sentence.

5 Conclusion

In this paper, we introduced our system submitted to WNGT2020 shared task to En→Ja track on STAPLE. We have shown that even a simple method which uses only forward and backward models' outputs can generate a variety of translations while maintaining original meaning and grammaticality.

In future, we plan to compare our system with existing systems that perform different types of language generation. In addition, we will investigate the impact of L2R and R2L models to the diverse output in depth.

References

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT^3: Web Inventory of Transcribed and Translated Talks. In *EAMT*.

Daphne Ippolito, Reno Kriz, Maria Kustikova, João Sedoc, and Chris Callison-Burch. 2019. Comparison of Diverse Decoding Methods from Conditional Language Models. In *ACL*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL*.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *LREC*.

Lemao Liu, Masao Utiyama, Andrew M. Finch, and Eiichiro Sumita. 2016. Agreement on Target-bidirectional Neural Machine Translation. In *HLT-NAACL*.

S. Mayhew, K. Bicknell, C. Brust, B. McDowell, W. Monroe, and B. Settles. 2020. Simultaneous Translation And Paraphrase for Language Education. In *WNGT@ACL*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *NAACL-HLT: Demonstrations*.

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

Raphael Shu, Hideki Nakayama, and Kyunghyun Cho. 2019. Generating Diverse Translations with Sentence Codes. In *ACL*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *CVPR*.

Jorg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *LREC*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NeurIPS*.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing He Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *AAAI*.

Qiongkai Xu, Juyan Zhang, Lizhen Qu, Lexing Xie, and Richard Nock. 2018. D-PAGE: Diverse Paraphrase Generation. *ArXiv*.