An Iterative Knowledge Transfer NMT System for WMT20 News Translation Task

Jiwan Kim, Soyoon Park, Sangha Kim, Yoonjung Choi
Individual Researchers
{jiwan.kim37,parksso0223,shkim000,choiyj35}@gmail.com

Abstract

This paper describes our submission to the WMT20 News translation shared task in English to Japanese direction. Our main approach is based on transferring knowledge of domain knowledge and linguistic characteristics by pre-training the encoder-decoder model with large amount of in-domain monolingual data through unsupervised and supervised prediction task. We then fine-tune the model with parallel data and in-domain synthetic data which is generated by iterative back-translation. For additional gain, we generate final results with an ensemble model and re-rank them with averaged models and language models. Through these methods, we achieve +5.42 BLEU score compared to the baseline model.

1 Introduction

This paper describes our submission to the WMT20 News translation task in English to Japanese direction. In this year, English-Japanese directions have newly established in News Translation Shared Task. The English-Japanese translation is not easy to deal with because of the difference in word order and the rich morphological characteristics of Japanese. Nevertheless, recent architectures for Neural Machine Translation (NMT), such as Transformer (Vaswani et al., 2017), show reasonable results when we have enough parallel data. Unfortunately, however, there is not much in-domain parallel data provided for English-Japanese task. To solve this issue, in this paper, we suggest the iterative knowledge transfer system which pre-trains the model with in-domain monolingual data.

Our system is based on Transformer architecture. We pre-train the model to transfer linguistic characteristics and domain knowledge of monolingual data. Although there are various pre-training methods for NMT, MASS (Song et al., 2019) is adopted in our system since MASS pre-trains the encoder and the decoder jointly and uses both labeled data and unlabeled data as the training data. To supplement insufficient in-domain parallel data, we generate synthetic data by back-translation from in-domain monolingual data. We also add some noise to the synthetic data. We then pre-train the model with the synthetic parallel data for supervised method and the monolingual data for unsupervised way. In fine-tuning step, we train the model with parallel corpus and perform the back-translation with in-domain data for iterative fine-tuning. In addition, we adopt an ensemble and averaging methods which are simple but very effective to improve performance in deep learning. With ensemble and average models, we apply noisy channel re-ranking which shows higher performance compared to R2L re-ranking (Yee et al., 2019). Through these methods, we achieve +5.42 BLEU score (Papineni et al., 2002; Post, 2018) compared to the baseline model.

2 Approach

Our system aims to encourage knowledge extraction of domain knowledge and linguistic characteristics by iteratively performing pre-training and fine-tuning. In this section, we explain techniques we use in each step.

2.1 Pre-training strategy

MASS is a masked sequence to sequence pre-training method for the encoder-decoder based language generation tasks (Song et al., 2019). The advantage of MASS is that it uses the encoder-decoder framework to predict the masked part given the masked sentence. Several consecutive tokens in a sentence are randomly masked; the encoder takes them as input, and the decoder is trained to predict masked tokens. This method allows MASS to learn the capability of representation

Proceedings of the 5th Conference on Machine Translation (WMT), pages 139–144
Online, November 19–20, 2020.

extraction. In this paper, we adopt both supervised and unsupervised prediction methods of MASS. There are plenty of in-domain monolingual corpus but insufficient in-domain parallel corpus. Thus, we generate synthetic data by back-translation and apply supervised prediction task. In addition, we use large amount of out-domain monolingual corpus for unsupervised prediction task to encourage the ability of language modeling.

Let $x \in \mathcal{X}$ as an monolingual source sentence, and m is the number of tokens of sentence x. We denote $x^{\setminus u:v}$ as an modified sentence of x where its position u to v are masked, $0 < u < v < m$. $x^{u:v}$ denotes the original sentence fragment of x from u to v. Those sentences can have different fragment positions u and v for each. In the sentence fragment, we replace each masked token to a special symbol $[\mathbb{M}]$, so the number of words in the sentence is not changed. Then, we train model with the masked sentence $x^{\setminus u:v}$ to predict the sentence fragment $x^{u:v}$. Supervised setting is used also where bilingual sentence pair $(x, y) \in (\mathcal{X}, \mathcal{Y})$ can be leveraged for pre-training. It is trained to predict y from the input $x^{\setminus u:v}$. The log likelihood in the entire setting is as follows:

$$
\begin{aligned}
L(\theta; (\mathcal{X}, \mathcal{Y})) &= \frac{1}{|\mathcal{Y}|} \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} \log P(y|x^{\setminus u:v}; \theta) \\
&+ \frac{1}{|\mathcal{X}|} \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} \log P(x|y^{\setminus u:v}; \theta) \\
&+ \frac{1}{|\mathcal{X}|} \sum_{x\in\mathcal{X}} \log P(x^{u:v}|x^{\setminus u:v}; \theta) \\
&+ \frac{1}{|\mathcal{Y}|} \sum_{y\in\mathcal{Y}} \log P(y^{u:v}|y^{\setminus u:v}; \theta)
\end{aligned} \tag{1}
$$

$P(y|x^{\setminus u:v}; \theta)$ and $P(x|y^{\setminus u:v}; \theta)$ denote the probability of translating a masked sequence to another language. This prediction task encourages the encoder to extract meaningful representations of masked input tokens in order to predict the unmasked output sequence.

2.2 Noised back-translation

Inspired from the noised back-translation (Edunov et al., 2018; Wu et al., 2019), we add noise to the train corpus. Let X and Y denote two languages, and let $\mathcal{X}$ and $\mathcal{Y}$ denote two corresponding sentence corpora, a set of all sentences. Let $\mathcal{B} = \{(x_i, y_i)_{i=1}^{N}\}$ denote the bilingual training corpus, where $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, and N is the number of sentence pairs. Let $\mathcal{M}_x = \{x_j\}_{j=1}^{N_x}$ and $\mathcal{M}_y = \{y_j\}_{j=1}^{N_y}$ denote sets of monolingual sentences, where N_x and N_y are sizes of each set, $x_j \in \mathcal{X}, y_j \in \mathcal{Y}$. We then train models $f_b : \mathcal{X} \mapsto \mathcal{Y}$ and $g_b : \mathcal{Y} \mapsto \mathcal{X}$ on the given bilingual data $\mathcal{B}$. Then, we build the following two synthetic datasets through the trained models:

$$
\begin{aligned}
\bar{\mathcal{B}}_{sx} &= \{(x, f_b(x))|x \in \mathcal{M}_x\}, \\
\bar{\mathcal{B}}_{sy} &= \{(y, g_b(y))|y \in \mathcal{M}_y\}, \\
\bar{\mathcal{B}}_{tx} &= \{(f_b(x), x)|x \in \mathcal{M}_x\}, \\
\bar{\mathcal{B}}_{ty} &= \{(g_b(y), y)|y \in \mathcal{M}_y\}
\end{aligned} \tag{2}
$$

where $\bar{\mathcal{B}}_{sx}$, $\bar{\mathcal{B}}_{sy}$ can be seen the forward translation of source-side monolingual data of X and Y and $\bar{\mathcal{B}}_{tx}$, $\bar{\mathcal{B}}_{ty}$ can be seen the backward translation of target-side monolingual data of X and Y.

We build following noise versions of the augmented datasets for training.

$$
\begin{aligned}
\bar{\mathcal{B}}^n_x &= \{(\sigma(x), \sigma(y))|(x, y) \in (\bar{\mathcal{B}}_{sx} \cup \bar{\mathcal{B}}_{ty})\}, \\
\bar{\mathcal{B}}^n_y &= \{(\sigma(y), \sigma(x))|(y, x) \in (\bar{\mathcal{B}}_{sy} \cup \bar{\mathcal{B}}_{tx})\}
\end{aligned} \tag{3}
$$

where $\sigma(x)$ denote the noised sentence of x, which consists of two types of noise: deleting tokens with probability 0.05 and swapping tokens in the sentence, implemented as a random permutation over the tokens with the uniform distribution but restricted to swapping words no further than three positions apart, where three is set empirically.

2.3 Noisy channel re-ranking

Noisy channel re-ranking method (Yee et al., 2019) is derived from Bayes' rule.

$$
p(y|x) = \frac{p(x|y)p(y)}{p(x)} \tag{4}
$$

Let x as a source sequence and y as a target sequence. Since $p(x)$ is constant for all y, only the channel model $p(x|y)$ and the language model $p(y)$ determine y when x is given. Score used for re-ranking can be calculated as follows:

$$
\frac{\alpha * logp(y|x) + \beta * logp(x|y) + \gamma * logp(y)}{|y|^p} \tag{5}
$$

where α, β, γ are tunable weight, and p is length penalty for target length $|y|$.

3 Experiments

3.1 Data

Data statistics The training data of the entire system is shown in Table 1. We use News Commentary (NC) data as another validation set in addition to newsdev2020 (devset).

Dataset	Lines
Parallel Data	
Wiki_Titles v2	0.7M
WikiMatrix	3.89M
Japanese-English Subtitle Corpus	2.8M
The Kyoto Free Translation Task	0.44M
TED Talks	0.24M
Monolingual Data (En)	
Europarl v10	2.29M
News Commentary v15	0.6M
News Crawl	23.35M
News Discussions	63.51M
Monolingual Data (Ja)	
News Crawl	3.44M
News Commentary v15	2983
Common Crawl	1773.97M

Table 1: Training corpora for our system

Preprocessing We use recaser in Moses (Koehn et al., 2007) to recase Japanese-English Subtitle Corpus where English side is lowercased. We also normalize punctuation marks and tokenize English corpus with Moses. We use Mecab (Kudo, 2006) to tokenize Japanese corpus. We adopt Sentencepiece (Kudo and Richardson, 2018); separate vocabs with 32K tokens are generated for each language. Separate vocabs show higher score in BLEU than a joint vocab in English-Japanese.

Filtering We first filter the parallel corpus based on length; sentences with more than 800 characters are removed from the training data. We then filter the training corpus with LangId (Lui and Baldwin, 2012). If LangIds of source or target side are mismatched, we filter out this data.

Data selection Unlike English, there are not enough news data in Japanese, so we select data from Common Crawl and use them as in-domain data. To obtain data close to in-domain, we classify sentences into in-domain and out-domain based on the perplexity of in-domain and out-domain language model (Moore and Lewis, 2010).

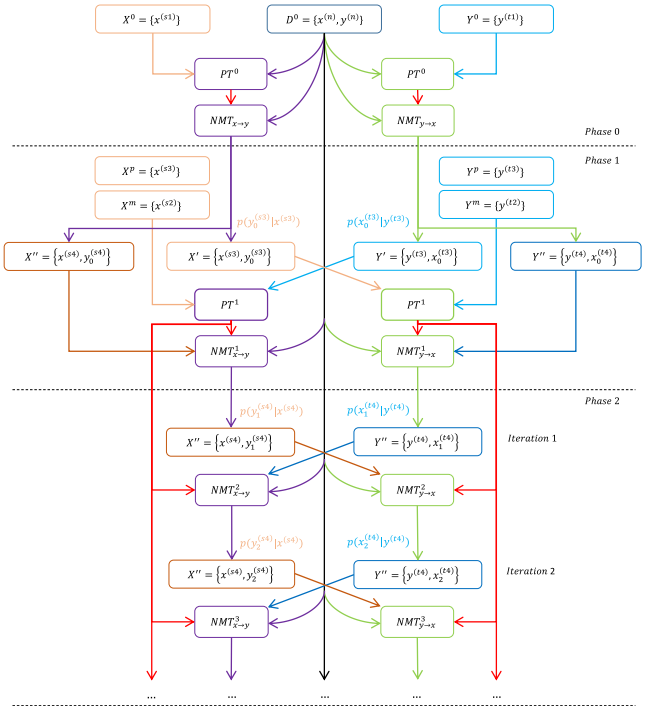

Figure 1: Illustration of training sequences of our system, where pre-trained models PT^* on both side are identical but separated for clarity.

Let $PPL_{in}(s)$ as the perplexity for sequence s with the in-domain language model and $PPL_{out}(s)$ as same with the out-domain language model. To classify sentences as close to in-domain, We calculate a score as follows:

$$\mathcal{S} = PPL_{out}(s) - PPL_{in}(s) \quad (6)$$

We train in-domain and out-domain language models respectively with KenLM (Heafield, 2011). The in-domain language model is trained with News Crawl corpus and the out-domain language model is trained with Common Crawl corpus.

3.2 Experimental setting

Our system is based on Transformer-big model on Fairseq (Ott et al., 2019)[1], which consists of 6-layers encoder and decoder each with 1024 embedding & hidden size and 4096 feed-forward layer size. Our system is trained using MASS[2] on 16×V100 GPUs, both in pre-training and fine-tuning.

3.3 Pre-training

Our entire training sequence is described in Figure 1. For the phase 0, we randomly sample 10M

[1] https://github.com/pytorch/fairseq
[2] https://github.com/microsoft/MASS

sentences X^0 and Y^0 from each mono corpus for unsupervised prediction task and use all available parallel corpus D^0 for supervised task. We prepare two separated prediction tasks, supervised and unsupervised setups respectively. For the supervised setup, we randomly mask entire input tokens in each sentence by 30% probability. In the unsupervised setup, we mask the fragment by replacing consecutive tokens with symbol $[\mathbb{M}]$ from random start position u. It first chooses 30% from input tokens, and each i-th token will be replaced as (1) an unchanged i-th token by 80% of the time, (2) a random token by 10% of the time, and (3) a masked token $[\mathbb{M}]$ by 10% of the time. After pre-training of model PT^0, two fine-tuned models $NMT_{x\to y}$ and $NMT_{y\to x}$ are trained with the parallel corpus, English-Japanese and Japanese-English direction respectively.

Lang	Lines	Remark
en	20M	
ja	20M	
ja*-en	5M	Randomly filtered
en*-ja	5M	LM-based filtered

Table 2: An amount of training corpora for pre-training. * means back-translated data from correspond monolingual corpus.

In the beginning of next phase, we create a new setup and train the model with training data mentioned in Table 2. We add noised synthetic data X' and Y' to create following version of training data. It consists of $\bar{\mathcal{B}}_{sx}$, $\bar{\mathcal{B}}_{sy}$, $\bar{\mathcal{B}}^n_x$ and $\bar{\mathcal{B}}^n_y$. X^m and Y^m consist of 20M mono corpora for unsupervised pre-training. 5M English mono corpus are randomly chosen from mono corpus, and 5M Japanese mono corpus are selected based on Equation 6; they are represented as X^p and Y^p in Figure 1. Then, 5M mono corpora are translated with $NMT_{x\to y}$ and $NMT_{y\to x}$ respectively.

PT^1 model is trained with above train corpus. Then, we train two fine-tuned models, $NMT^1_{x\to y}$ and $NMT^1_{y\to x}$ separately with parallel corpora in Table 3.

3.4 Iterative fine-tuning

After pre-training in phase 1, we create fine-tuned models with parallel corpus D^0 and synthetic corpus X'' and Y''.

Inspired from joint training (Zhang et al., 2018), we perform back-translation and fine-tune steps iteratively in phase 2. Synthetic corpora for each steps are replaced to a newly generated ones from developed models, which are represented as X'' and Y'' in Figure 1.

Lang	Lines	Domain	Remark
			English - Japanese
en-ja	7M	out	
en*-ja	3M	in	
			Japanese - English
ja-en	7M	out	
ja*-en	7M	in	Randomly filtered

Table 3: An amount of training corpora for fine-tuning

3.5 Advance decoding

We improve our final result with noisy channel re-ranking method (Yee et al., 2019). The small difference is we use the different direct model for scoring instead of using the same model used for generation. To generate y, we first ensemble three models with final back-translated models, considering validation sets. We generate 44 n-bests results with 44 beam size with ensemble models. Then, we re-rank the results according to Equation 5. The direct model for scoring is the averaged model of three models used for ensemble. This is faster and shows better results compared to the ensemble model. The channel model is an average model in the opposite direction. For language model, we use Transformer-big model, trained only with News domain monolingual corpus. Finally, we tune weights of each model and length penalty with validation sets.

3.6 Experimental Results

Step	Model	Dev
	Baseline	17.62
Phase 0	MASS	19.16
Phase 1	MASS	19.23
	+ Noise	19.31
Phase 2	Back-translation Iter1	23.59
	Back-translation Iter2	23.91
	Ensemble	24.05
	+ Beam 44	24.21
	Re-ranking(devset)	24.73
	Re-ranking(NC)	23.55

Table 4: En-Ja BLEU scores on WMT20 devset

Model	Test
Baseline	20.51
Ensemble + Beam 44	25.05
Re-ranking(devset)	24.41
Re-ranking(NC)	**25.93**

Table 5: En-Ja BLEU scores on WMT20 test set.

The results of English to Japanese direction are shown in Table 4 and 5. Our final submission's BLEU score is 5.42 higher than the baseline model.

For evaluation, multi-bleu.perl[3] is used after tokenizing with Mecab in Japanese. The baseline model is trained only with parallel data in Transformer-big architecture and is decoded with beam size 4. It shows great performance improvement when MASS is applied. When using synthetic data and adding noise to data in pre-training steps (Phase 1), it shows better results compared to it with only parallel data (Phase 0). Back-translation with the in-domain monolingual data increases the BLEU score most, and the score increases further in the next iteration. The ensemble model and large beam size also show better BLEU score.

For the test set, we replace symbol £ to "pound" in source sentences as pre-processing. We re-rank and tune the parameters based on News Commentary parallel data set which shows better results than tuning with devset. Since we select best models based on devset in previous steps, using devset in re-ranking seems to result in overfitting.

The final result of our submission is shown in Table 6. Characters based tokenizer and SacreBLEU[4] are used for evaluation in Ocelot.

Submission	SacreBLEU	chrF
English-Japanese	41.0	0.351

Table 6: Automatic evaluation on WMT20 test set in Ocelot.

4 Conclusions

In this paper, we describe our submission to the WMT20 news translation task in English to Japanese direction. Our main approach is based on transferring knowledge from large amount of monolingual data by pre-training the model iteratively using MASS. We then improve the system with several effective methods: noised and iterative back-translation, in-domain data selection, and re-ranking. Through these methods, we achieve competitive results compared to the baseline and prove that the iterative knowledge transfer system we proposed is effective.

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

[4] https://github.com/mjpost/sacrebleu

References

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *CoRR*, abs/1808.09381.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2006. Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab. sourceforge. jp*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

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 *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation, Volume 1: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216, Hong Kong, China. Association for Computational Linguistics.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. *CoRR*, abs/1803.00353.