A Study on Exploiting Additional Resources for Low-resource Neural Machine Translation

Aizhan Imankulova

Department of Information and Communication Systems
Graduate School of System Design
Tokyo Metropolitan University

March 2021

A Doctoral Dissertation
submitted to Graduate School of System Design,
Tokyo Metropolitan University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Aizhan Imankulova

Thesis Committee:
Mamoru Komachi (Associate Professor, Tokyo Metropolitan University)
Toru Yamaguchi (Professor, Tokyo Metropolitan University)
Yasufumi Takama (Professor, Tokyo Metropolitan University)
Katsuhito Sudoh (Associate Professor, Nara Institute of Science and Technology)

Dedication

To my late parents, Raushan and Toktasyn, whose kindness, love and support made me who I am today. I will always remember you...

Acknowledgements

This thesis would not have been possible without the contribution of many people who, throughout the years, not only influenced my work but also supported me in many ways during this adventure.

I am immensely grateful to my supervisor Professor Mamoru Komachi for welcoming me to the Komachi laboratory and giving me guidance, comfort, and encouragement. I will always be grateful to you for supporting me in every aspect of my student life, starting from the moment I contacted you from Kazakhstan till the very end, especially when I had hard times with my research life. Thank you for letting me freely choose the research themes I wanted to explore and encouraging me to go out and be able to have work with many researchers and gain a lot of experience.

I am very grateful to Atsushi Fujita, Kenji Imamura, and Raj Dabre, from the National Institute of Information and Communications Technology (NICT). Thanks to their guidance, I learned methods of researching, such as breaking down research subjects into small steps and tackling them by putting a lot of thought into each step. Also, you taught me the importance of making many mistakes as soon as possible and learning from them. And I learned the importance of explaining motivations and conclusions first and explaining them in an easy-to-understand manner. And most importantly, I learned about the importance of defending my work.

I would like to express my warm thanks to my mentors from the Rakuten Institute of Technology (RIT) to Koji Murakami, Lasguido Nio, Yo Hirate, and Vijay Daultani. In RIT, I have experienced how important machine translation technology in companies and the real world. And I learned many things that I could not experience at university, such as researchers' role in companies.

My special thanks go to my thesis committee members: Professors Toru Yamaguchi, Yasufumi Takama, Katsuhito Sudoh for reviewing this thesis. I was able to complete this thesis thanks to your insightful advice, comments, and detailed feedback.

I would also like to thank Dr. Tomoyuki Kajiwara, for his strict and yet very kind guidance, especially at the beginning of my researcher path. I learned from you the importance of explaining and writing complicated matters in an easy-to-understand manner. Even after graduation, I was concerned about it and asked for various consultations. I am grateful that even after you graduated, you were concerned about me and consulted me about various topics.

A great thank you goes to Yui Suzuki for always supporting me as a mentor to become familiar with Japan and the laboratory. I was delighted that we became friends in research and personal life, and I will dearly remember the welcome party for me and other gatherings at your home with delicious food and interesting games.

Many thanks to my colleagues and friends from the Komachi laboratory, to Masahiro Kaneko, Tosho Hirasawa, Takayuki Sato, Siti Oryza Khairunnisa, Ikumi Yamashita, Satoru Katsumata, Zhousi Chen, Longtu Zhang for always helping me and supporting me.

I would also like to show my gratitude to Rob van der Goot, Desmond Elliot, Ozan Caglayan, and Emanuele Bugliarello for their expertise, support, insightful ideas, comments, valuable feedback, and interesting discussions.

A very special thank you goes to Jun Kawai, Tomomi Kawai, and Carlos Salazar for supporting my student life, which can be difficult, in my personal life.

Finally, I would like to thank my families and friends in Kazakhstan and Japan, aunt Roza, Nurgul, Roka, Miwa, Masanobu, and many-many others, for their unending support and belief in me.

Abstract

Machine translation (MT) is the task of translating input text from a source language into a target language. The practical use of MT will enable smooth communication between different languages. In the real world, the MT research results are applied as various services such as Google translation and DeepL. One of the breakthroughs in MT in recent years is the arrival of neural machine translation (NMT). NMT models have been reporting significant performance improvements. On the other hand, neural MT models require a large number of parallel sentences for training.

The biggest issue with low-resource languages is the extreme difficulty of obtaining enough resources. MT has proven successful for several language pairs. However, each language comes with its challenges. Low-resource languages have largely been left out of the MT revolution. For instance, there are often very few written texts, and even the languages that have monolingual text do not always have a parallel text in another language.

We research to what extent it is possible to improve MT systems' performance in a low-resource scenario using other pseudo-parallel data, other helping language pairs, and other modality data to increase the training data size for different language pairs and domains.

Previously, additional training data has been augmented by pseudo-parallel corpora obtained by using MT models to translate monolingual corpora into the source language. However, in low-resource language pairs, in which only low accuracy MT systems can be used, translation quality degrades when a pseudo-parallel corpus is naively used. Therefore, we consider data selection and filtering of the generated pseudo-parallel corpora using different similarity metrics.

Another way to improve low-resource MT would be to use out-of-domain data. However, merely using MT systems trained on out-of-domain data for in-domain translation is known to perform poorly. To effectively use large-scale out-of-domain data for low-resource tasks, we need to utilize domain adaptation and multilingual transfer approaches. In order to do that, we propose a multistage fine-tuning method, which combines two types of transfer learning, i.e., domain adaptation and multilingual transfer

from other language pairs with conventional fine-tuning, where an NMT system trained on out-of-domain data is fine-tuned only on in-domain data, or mixed fine-tuning, where pre-trained out-of-domain NMT system is fine-tuned using a mixture of in-domain and out-of-domain data.

Different from conventional full-sentence MT, simultaneous MT is also considered to be one of the low-resource scenarios due to involving translating a sentence before the speaker's utterance is completed in order to realize real-time understanding. This task is significantly more laborious than the general full sentence translation because of the shortage of input information during decoding. To alleviate this shortage, we propose to leverage visual clues as an additional modality to help MT systems predict translations from richer information.

The main contribution of this thesis is improving MT performance for low-resource language pairs by effectively using additional information from different resources. To improve MT performance with low-resource language pairs, we propose methods to effectively expand the training data via filtering the pseudo-parallel corpus based on back-translation and round-trip translation. Furthermore, we propose a novel multilingual multistage fine-tuning approach for low-resource NMT, taking a challenging Japanese–Russian pair for benchmarking.

By using additional modality to simultaneous MT, we verified the importance of visual information during decoding by performing throughout the evaluation and analyzing its effect on different low-resource language pairs.

This thesis is organized as follows:

Chapter 1 gives an overview of MT and its challenges in low-resource scenario. It describes aim and objectives of improving MT performance for low-resource language pairs by effectively using additional information from different resources.

Chapter 2 introduces methods of creating and filtering pseudo-parallel corpora by back-translation and round-trip translation. Here, I show that using filtered pseudo-parallel corpora as additional training data improves NMT performance compared to using unfiltered pseudo-parallel corpora for both back-translation and round-trip translation methods. The proposed method achieved up to 3.46 BLEU points in the Russian→Japanese translation and up to 5.25 BLEU points in the Japanese→Russian translation.

Chapter 3 addresses the research questions of the advantages and disadvantages of out-of-domain data for low-resource language pairs. To effectively exploit out-of-domain parallel data, I propose a multistage fine-tuning method, which combines domain adaptation multi-lingual transfer approaches. The proposed method achieved up to 2.72 BLEU points in the Russian→Japanese and up to 3.06 BLEU points in the Japanese→Russian translation.

Chapter 4 introduces a novel technique of utilizing different modality for low-resource simultaneous MT. In this chapter, I propose to combine multimodal and simultaneous NMT to enrich incomplete text input information using a visual clue. As a result, the proposed method significantly outperformed text-only baselines in all experimented language-pairs, especially for language pair with different word orders such as English→Japanese.

Chapter 5 concludes this thesis, discusses insights and limitations, and describes potential future work for low-resource MT.

Keywords— Natural language processing, machine translation, low-resource, data sparsity, additional information, additional modality, pseudo-parallel corpus, filtering, multistage fine-tuning, visual information

Contents

List of Figures

List of Tables

1 | Introduction

1.1 Machine Translation

Since we live in an increasingly connected world, translation became an essential tool, allowing us to connect and share information, no matter what the language. However, translating vast amounts of content could bring complications around cost, quality, and time. Therefore, machines have come to our help in order to remedy some of these potential issues.

Machine Translation (MT) is the field of Natural Language Processing (NLP) which aims to translate text from one language to another. MT is used to improve the capacity of translation, allowing for more content to be translated by reducing financial, human, and time costs.

Currently there are two major approaches: Phrase Based Statistical Machine Translation (PBSMT) (Koehn et al., 2003) and Neural Machine Translation (NMT) (Bahdanau et al., 2014; Cho et al., 2014; Sutskever et al., 2014; Luong and Manning, 2015). Both approaches rely on parallel text corpora, which contain source language texts $\mathbf{X} = (x_1, ..., x_n)$ of length n and their translations $\mathbf{Y} = (y_1, ..., y_t)$ of length t in target language.

PBSMT tries to learn a probabilistic model from data using Bayes rule:

$$\text{argmax}_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}) = \text{argmax}_{\mathbf{Y}} P(\mathbf{X}|\mathbf{Y}) P(\mathbf{Y}) \tag{1.1}$$

Here, $P(\mathbf{X}|\mathbf{Y})$ is called the translation model, trained on parallel corpus. $P(\mathbf{Y})$ is called the language model, trained on monolingual target language corpus only.

NMT is an end to end deep learning approach which mainly uses a single neural network architecture. Most NMT models are deep consisting of several layers of neurons to process input sequences. Opposed to PBSMT, it directly calculates $P(\mathbf{Y}|\mathbf{X})$:

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{|\mathbf{Y}|} P(y_t|\mathbf{X}, y_{<t}) \tag{1.2}$$

Conventionally, translation is performed after receiving full input text sequence, as known as full-sentence MT. However, one can achieve the translation in a simultaneous way, as known as simultaneous MT, which starts translation before the full input sequence is received (Matsubara et al., 2000; Grissom II et al., 2014; Gu et al., 2017; Ma et al., 2019). Also, one can operate not only with text modality but also use other modalities, such as image, video, or speech, as inputs to improve MT accuracy, as known as multimodal MT (Elliott et al., 2016; Specia et al., 2016; Elliott et al., 2017).

1.2 Evaluation Methods

Progress of machine translation relies on assessing a new system's quality to show that the new system can perform better than previous systems. However, human evaluation is also a very costly activity similar to human translation, especially considering how fast new systems and their intermediate versions are created and tested. One may want to evaluate tens or hundreds of systems a day, for example, to find the best model within models created at each epoch of training or to find the best hyper-parameters that lead to better MT models. Therefore, it is crucial to find automatic MT evaluation metrics since performing evaluation manually is not remotely feasible (Papineni et al., 2002). In the following sections, we will describe some MT evaluation metrics used in this work.

1.2.1 BLEU

Bilingual Evaluation Understudy Score (BLEU) (Lin and Och, 2004a,b) is the current widespread standard for automatic MT evaluation, mostly because it is quick, inexpensive, and language-independent. The BLEU score of system output is calculated by counting the number of n-grams in the system output, matched with the set of n-grams in reference translations. The highest n-gram order is defined commonly to be four. Precision is calculated separately for each n-gram order, and the precisions are combined via a geometric averaging as follows:

$$p_n = \frac{\sum_{c \in C_n} Count_{clip}(c)}{\sum_{c' \in C'_n} Count(c)} \tag{1.3}$$

Here, C_n indicates a set of n-grams, $Count_{clip}$ truncates each word's count, if necessary, not to exceed the largest count observed in any single reference for that word.

As a result BLEU score is calculated as follows:

$$\text{BLEU} = \exp(\sum_{n=1}^{N} w_n \log p_n) \tag{1.4}$$

Here, w_n denotes positive weights summing to one. The result is typically measured on a 0 to 1 scale, with 1 as the hypothetical ”perfect” translation. Since the human reference, against which MT is measured, is always made up of multiple translations, even a human translation would not score a 1. In this work, we express BLEU by multiplying it by 100.

1.2.2 RIT score

This evaluation metric is specific to the e-commerce domain. RIT score (Murakami et al., 2017) uses external vocabulary such as product information registered in a United States E-commerce company Rakuten.com[1] to calculate how many words have appeared in the equivalent categories. It is based on the hypothesis that English words appearing on the Rakuten.com site are necessary words to sell products of that category. Therefore, the sentence-level RIT score (RIT score) is a weighted sum of precision of all correct n-grams in the external vocabulary. It was calculated as:

$$\text{RIT score} = 4p_1 + 9p_2 + 3p_3. \tag{1.5}$$

[1] https://www.rakuten.com

1.2.3 Average Lagging

Average Lagging (AL) latency metric that evaluates simultaneous MT systems' outputs, which was proposed by Ma et al. (2019). [2] It calculates the degree of out of sync time with the input, in terms of the number of source tokens as follows:

$$\mathrm{AL}_g(\mathbf{X}, \mathbf{Y}) = \frac{1}{\tau_g(|\mathbf{X}|)} \sum_{t=1}^{\tau_g(|\mathbf{X}|)} g(t) - \frac{t-1}{r} \tag{1.6}$$

where $r = |\mathbf{Y}|/|\mathbf{X}|$ is the target-to-source length ratio and τ_g is the decoding step when source sentence finishes:

$$\tau_g(|\mathbf{X}|) = \min\{t|g(t) = |\mathbf{X}|\} \tag{1.7}$$

1.3 Challenges in Low-resource Machine Translation

Large-scale parallel corpora are essential for training high-quality machine translation systems; however, such corpora are not freely available for many language translation pairs.

In this research, we focus on low-resource machine translation. We define two aspects of low-resource machine translation:

- Limited availability of data during training. This aspect represents situations where we have only 10,000-30,000 parallel sentences for the language pair of interest, such as Japanese↔Russian, in order to train desired MT systems. This problem can also relate to broadly known high-resource language pairs, such as English↔French, English↔Russian, English↔Chinese, because of the limited availability of parallel data in some specific domains, such as e-commerce, news, spoken domains, etc. One of the solutions is to use additional resources, such as monolingual data or out-of-domain data.

- Limited availability of data during translation. This aspect represents situations, where given input information is not sufficient in order to generate the translation accurately. This problem can be related to all languages and

[2] https://github.com/SimulTrans-demo/STACL

domains in such scenarios as simultaneous MT or MT of noisy input. One of the solutions is to use information from an additional modality.

In the following sections, we will describe the challenges of using the mentioned additional resources.

1.3.1 Monolingual Corpora

A large-scale parallel corpus is an essential resource for training PBSMT and NMT systems. Creating a high-quality, large-scale parallel corpus requires time, financial resources, and expert translation of a large amount of text. Resultingly, many existing large-scale parallel corpora are limited to specific languages and domains. Contrastingly, large monolingual corpora are easier to obtain.

Various approaches have been proposed to create a pseudo-parallel corpus from a monolingual corpus. For example, Zhang and Zong (2016) proposed a method to generate a pseudo-parallel corpus based on a monolingual corpus of the source language and its automatic translation. Sennrich et al. (2016a) obtained substantial improvements by automatically translating a monolingual corpus of the target language into the source language, referred to as synthetic source sentences, and treating the obtained pseudo-parallel corpus as additional training data. They used monolingual data of the target language to learn the conditional language model more effectively. However, they experimented on language pairs for which relatively large-scale parallel corpora are available. Thus, they did not fully exploit the training corpus or address the quality of the pseudo-parallel corpus.

The pseudo-parallel corpus quality is critical because low-quality parallel sentences will degrade NMT performance more than SMT (Koehn and Knowles, 2017). Accordingly, our motivation is to filter out low-quality synthetic sentences that might be included in such a pseudo-parallel corpus to obtain a high-quality pseudo-parallel corpus for low-resource language pairs.

1.3.2 Out-of-Domain Corpora

Another way to improve low-resource MT would be to use out-of-domain data. However, simply using MT systems trained on out-of-domain data for in-domain translation is known to perform poorly (Haddow and Koehn, 2012; Koehn and Knowles, 2017). For example, the conventional method is fine-tuning, in which a model trained on out-of-domain data is further trained on in-domain data (Luong and Manning, 2015; Chu et al., 2017). However, fine-tuning tends to overfit quickly due to the small size of the in-domain data. Another challenge appears when there is not any available out-of-domain data for the language pair of interest. Therefore, there is a need to use out-of-domain data from other language pairs.

To effectively use large-scale out-of-domain data for low-resource tasks, we need to utilize domain adaptation, and multilingual transfer approaches in multiple stages.

1.3.3 Additional modality

Unlike conventional full-sentence MT, simultaneous MT is considered one of the low-resource scenarios due to involving translation of a sentence before the speaker's utterance is completed to realize real-time understanding. It is widely used in international summits and conferences where real-time comprehension is one of the most important aspects. Simultaneous translation is already a difficult task for human interpreters because the message must be understood and translated while the input sentence is still incomplete, especially for language pairs with different word orders (e.g., SVO-SOV) (Seeber, 2015). Consequently, simultaneous translation is more challenging for machines. Previous works attempt to solve this task by predicting the sentence-final verb (Grissom II et al., 2014), or predicting unseen syntactic constituents (Oda et al., 2015). Given the difficulty of predicting future inputs based on existing limited inputs, Ma et al. (2019) proposed a simple simultaneous neural machine translation (SNMT) approach `wait-k` which generates the target sentence concurrently with the source sentence, but always `k` tokens behind, satisfying low latency requirements.

Simultaneous interpreters often consider various additional information sources such as visual clues or acoustic data while translating (Seeber, 2015). Therefore, we hypothesize that using supplementary information, such as visual clues, can also be beneficial for simultaneous MT.

However, previous approaches solve the given task by solely using the text modality, which may be insufficient to produce a reliable translation. To alleviate this shortage, we propose to leverage visual information as an additional modality to help simultaneous MT systems predict translations from richer information, despite the fact that the improvement brought by visual features for full sentence MT is moderate (Hitschler et al., 2016; Specia et al., 2016; Elliott and Kádár, 2017).

1.4 Aim and Objectives

The aim of this thesis is to improve machine translation performance for low-resource language pairs by effectively using additional information from different resources.

To improve machine translation performance with low-resource language pairs, we propose methods to effectively expand the training data via filtering the pseudo-parallel corpus based on back-translation and round-trip translation. Furthermore, we propose a novel multilingual multistage fine-tuning approach for low-resource neural MT (NMT), taking a challenging Japanese–Russian pair for benchmarking.

By using an additional modality for simultaneous MT, we verified the importance of visual information during decoding by performing throughout the evaluation and analyzing its effect on different low-resource language pairs.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2 — introduces methods of creating and filtering pseudo-parallel corpora by back-translation and round-trip translation. Here, we show that using filtered pseudo-parallel corpora as additional training data improves NMT performance compared to using unfiltered pseudo-parallel corpora for both back-translation and round-trip translation methods. The proposed method achieved up to 3.46 BLEU points in the Russian→Japanese translation and up to 5.25 BLEU points in the Japanese→Russian translation.

Chapter 3 — addresses the research questions of the advantages and disadvantages of out-of-domain data for low-resource language pairs. To effectively exploit out-of-domain parallel data, we propose a multistage fine-tuning method, which combines domain adaptation multilingual transfer approaches. The proposed method achieved up to 2.72 BLEU points in the Russian→Japanese and up to 3.06 BLEU points in the Japanese→Russian translation.

Chapter 4 — introduces a novel technique of utilizing different modality for low-resource simultaneous MT. This chapter proposes to combine multimodal and simultaneous NMT to enrich incomplete text input information using a visual clue. As a result, the proposed method significantly outperformed text-only baselines in all experimented language-pairs, especially for language pair with different word orders such as English→Japanese.

Chapter 5 — concludes this thesis, discusses insights and limitations, and describes potential future work for low-resource MT.

2 | Pseudo-Parallel Corpora for Low-Resource NMT

2.1 Introduction

In this chapter, we propose two ways of creating and **filtering** pseudo-parallel corpora. The proposed methods involve filtering a pseudo-parallel corpus by (1) back-translation and (2) round-trip translation of a monolingual corpus for low-resource language pairs.

The main contributions are as follows:

- To establish a high-quality pseudo-parallel corpus, we filter a pseudo-parallel corpus generated by round-trip translation using three sentence-level similarity metrics: sentence-level Bilingual Evaluation Understudy Score (sent-BLEU) (Lin and Och, 2004a,b), average alignment similarity (AAS), and maximum alignment similarity (MAS) (Song and Roth, 2015). We also use a sentence-level language model (sent-LM) and RIT score (Murakami et al., 2017) to filter a pseudo-parallel corpus generated by back-translations and only considering synthetic source sentences.
- We observe that bootstrapping using our proposed filtering method significantly improves the BLEU score; however, the gains in BLEU decrease gradually over several iterations.
- We show that the proposed filtering method, along with bootstrapping, is useful for low-resource language pairs.
- We publicly released the obtained filtered pseudo-parallel corpora generated by round-trip translation.[1]

[1] https://github.com/aizhanti/filtered-pseudo-parallel-corpora

2.2 Related Work

To address data sparsity in machine translation, many methods use monolingual data to improve translation quality (Ueffing et al., 2007; Schwenk, 2008; Bertoldi and Federico, 2009; Hsieh et al., 2013; Zhang and Zong, 2016; Zhang et al., 2018; Edunov et al., 2018). Specifically, Bertoldi and Federico (2009) addressed the problem of domain adaptation by training a translation model from a generated pseudo-parallel corpus created from a monolingual in-domain corpus. Hsieh et al. (2013) created a pseudo-parallel corpus from patterns learned from source and monolingual target in-domain corpora for cross-domain adaptation. They manually conducted filtration of "relatively more accurate" translated sentences and used them to revise the language model. Several methods use iterative approaches to improve NMT using pseudo-parallel corpora (Hoang et al., 2018; Cotterell and Kreutzer, 2018). Zhang et al. (2018) used both source and target monolingual corpora to improve an NMT system iteratively. Edunov et al. (2018) improved high-resource NMT using synthetic sources generated by sampling or adding noise to beam outputs. Imamura et al. (2018) sampled multiple sources for each target sentence to enhance the encoder and attention mechanism, leading to an improvement of translation quality. However, experiments were conducted on relatively high-resource language pairs. Cheng et al. (2016) presented a semi-supervised approach to training bidirectional neural machine translation models using autoencoders on the monolingual corpora with high-resource source-to-target and target-to-source translation models as encoders and decoders. Their settings are different from ours in terms of (1) the available size of parallel data to train their round-trip translation models and (2) using created pseudo-parallel data as is without filtration. Niu et al. (2018) improved bi-directional NMT by continuously training on augmented parallel data. Similarly, in this study, we used a pseudo-parallel corpus created by translating a monolingual corpus from the target language rather than from the source language. Contrastingly, automatic filtering is applied to the obtained pseudo-parallel corpus. We conducted experiments on low-, medium- and high-resource language pairs to demonstrate the accuracy of the filtered pseudo-parallel corpora created by the NMT.

Data filtering is often used in domain adaptation (Moore and Lewis, 2010; Axelrod et al., 2011) for phrase-based SMT systems. Sentences are extracted from large corpora to optimize the language model and the translation model (Wang et al., 2014; Yıldız et al., 2014). The work most closely related to our study is that of Yıldız et al. (2014), in which a quality estimator was built to obtain high-quality parallel sentence pairs using a bilingual dictionary. They achieved improved translation performance and reduced the time complexity with a small high-quality corpus. This method filters data by calculating the similarity between the source and target sentences. The similarity is calculated between monolingual and synthetic target sentences without relying on any external dictionaries in our work.

Recently, van der Wees et al. (2017) performed dynamic data selection in the training of an NMT model. To sort and filter the training data, they used language models from the source and target sides of in-domain and out-of-domain data to calculate cross-entropy scores. However, in the present study, round-trip translation is employed to filter data while taking into consideration their meaning.

Meanwhile, He et al. (2016a) presented a dual-learning approach. It simultaneously trains two models through a reinforcement learning process. Monolingual data of both source and target languages are used, and informative feedback signals are generated to train the translation models. The dual-learning approach was shown to alleviate the issue of noisy data by increasing its quality. In our approach, on the other hand, we attempt to remove noisy data. In addition, He et al. (2016a) assumed a high-resource language pair to "warm-start" the reinforcement learning process, whereas we target low-resource language pairs, wherein high-quality seed NMT models are difficult to obtain.

A completely unsupervised approach (Artetxe et al., 2018; Lample et al., 2018a) has been useful in a zero-shot scenario by exploiting only monolingual corpora and back-translation. Nevertheless, we focus on maximizing the utility of existing small parallel corpora, leaving the application of recent unsupervised MT methods for future work.

2.3 Pseudo-Parallel Corpora by Back-Translation

In this section, we investigate the effect of pseudo-parallel corpora created and filtered by a back-translation approach using RIT score (Murakami et al., 2017). Here, we deal with challenging Japanese→English e-commerce product titles corpus, which contains noisy translations and a different set of vocabulary for each e-commerce product category. Data in each category is low-resource and has different domains.

E-commerce product sales are dramatically rising around the world, so are selling e-commerce products abroad. However, to successfully sell e-commerce products abroad, overcoming the language barrier becomes one of the important steps. Therefore, machine translation could be a solution in translating a great amount of e-commerce product texts.

E-commerce product data are different from those used in academia. Below are the differences between Rakuten Ichiba Japanese→English parallel e-commerce product titles and academia-wise data: 1) E-commerce product data were created by individual stores that have translated and registered text about their products. However, some Japanese sentences were translated using machine translation tools without any proofreading. This resulted in low-quality, noisy parallel data with mistranslated proper names and erroneous grammar, which required transcreation. 2) The amount of created parallel data in Rakuten Ichiba is small in comparison to the existing Japanese monolingual e-commerce product data. 3) E-commerce product data are divided into many categories, and each category has a different set of vocabulary. This kind of parallel data includes a wide range of products, proper names, and descriptions, which leads to the data sparseness problem with too many unique word types, especially in NMT. All of these factors adversely affect the performance of a low-resource NMT.

We compare several Japanese→English corpora which are used in academia (NTCIR, ASPEC, Tanaka) to train NMT systems and Rakuten Ichiba parallel e-commerce product titles corpus (Rakuten), which is a concatenation of all available Rakuten Ichiba parallel data from different categories. Table 2.1 shows the ratio of the number of unique word types to the number of total tokens in the Japanese→English parallel corpora. The higher the ratio, the more difficult to train an NMT model using a given parallel corpus.

Under these restricted conditions, it is necessary to consider how NMT should be

Corpus	# of sent	Ratio for Ja	Ratio for En
NTCIR	1,169,201	0.002	0.005
ASPEC	3,008,500	0.004	0.011
Tanaka	148,835	0.018	0.016
Rakuten	1,228,207	0.023	0.025

Table 2.1: Ratio for Japanese→English parallel corpora.

applied to translate e-commerce product titles. Therefore, we first investigate how to handle low-frequency words in e-commerce product data; and if one general NMT model could be enough for all e-commerce categories or if we should train an NMT model for each category. Then we propose to select data with better quality from given parallel data to train an NMT model for e-commerce products using RIT score. Next, we show the effect of data augmentation by back-translating and filtering those pseudo-parallel data using RIT score in order to further improve NMT for e-commerce product titles (Figure 2.1).

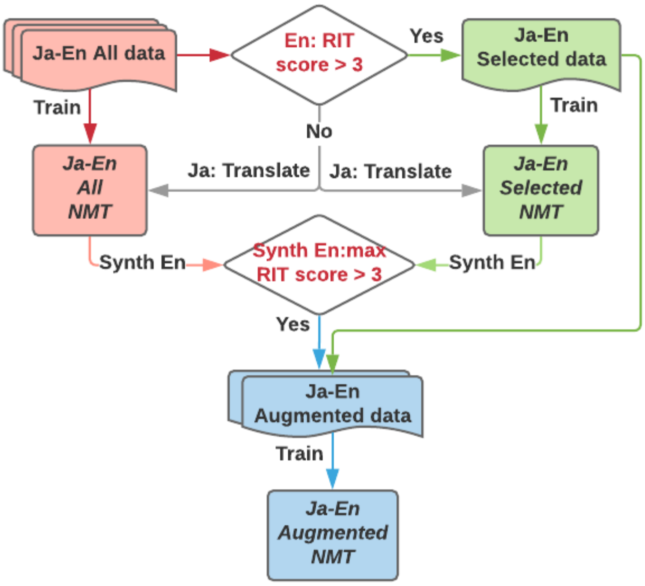


Figure 2.1: Flow of creating pseudo-parallel corpus by back-translation and filtering by RIT score.

2.3.1 Experimental Setup

Translation Model

In this section, for all experiments, we translate from Japanese into English.

We used an open source OpenNMT toolkit[2] described by Klein et al. (2017) for experiments. We used recommended methods by Denkowski and Neubig (2017) such as Byte Pair Encoding (BPE) (Sennrich et al., 2016b) and annealing Adam optimization (Kingma and Ba, 2015). Adam has a maximum step size of 0.0002. A bi-directional encoder and decoder with a single LSTM layer have 1,024 units, and for word representations, we used 512 units.

For evaluation we used BLEU (Papineni et al., 2002) and RIT scoring system (Murakami et al., 2017).

We tokenized English sentences using the NLTK script and removed non-letter characters. For Japanese sentences, we removed meta-tags and used MeCab 0.996 with the mecab-ipadic-NEologd[3] dictionary for word segmentation. We eliminated the sentence pairs exceeding 50-word length from all data, with length ratio bigger than 3 and duplicated pairs. For data selection, we used the RIT score.

Dataset

We experimented with Japanese→English translation using in-house e-commerce product titles, which are spread across different categories with different size. We chose 9 separate categories to experiment with: Breadmaker, Microwave, Pendant, Rice cooker, Shampoo, Shoes, Skirt, Socks, and Tops. In our case, data of the Rakuten e-commerce product titles have a tree structure: each leaf represents one category; nodes represent combined categories (Table 2.2, Tools and Clothes), and root contains data from all listed categories (Table 2.2, Rakuten). Tools contain data from Breadmaker, Microwave, Pendant, Rice cooker, and Shampoo. Clothes contain data from Shoes, Skirt, Socks, and Tops. These make up for the 12 datasets we experimented on. For development and test sets, we randomly sampled sentences from each dataset that have RIT score ≥ 3 (for Breadmaker, which has little

[2] http://opennmt.net

[3] https://github.com/neologd/mecab-ipadic-neologd

data, we set RIT score $\geq$ 2.5) in order to calculate BLEU scores on more reliable data. Table 2.2 shows the data statistics after preprocessing for 12 datasets, which we used in our experiments.

Dataset	# of sentence pairs			Word frequency for Japanese			Word frequency for English		
	Train	Dev	Test	# of types	1	2	# of types	1	2
Breadmaker	500	91	100	1,076	423	171	1,091	535	146
Microwave	1,065	199	200	1,825	704	257	1,948	905	271
Pendant	140,390	1,000	1,000	72,193	39,409	9,811	60,392	35,238	7,595
Rice cooker	3,049	600	600	3,480	1,461	594	4,370	2,246	787
Shampoo	32,368	1,000	1,000	14,731	5,434	2,089	11,901	4,640	1,621
Tools	177,372	2,890	2,900	85,089	44,190	11,586	71,337	40,423	9,078
Shoes	414,316	2,000	2,000	192,080	87,571	30,077	185,211	100,383	28,996
Skirt	52,007	1,000	1,000	48,544	29,030	6,263	36,955	22,793	4,134
Socks	34,356	1,000	1,000	25,748	11,707	3,958	20,698	9,787	3,035
Tops	550,156	2,000	2,000	264,140	148,776	35,502	239,347	158,349	27,388
Clothes	1,050,835	6,000	6,000	427,737	231,182	60,239	411,161	259,819	53,307
Rakuten	1,228,207	8,890	8,900	479,034	259,430	67,345	457,342	289,244	59,113

Table 2.2: Data statistics for 12 category. Columns “1” and “2” show the number of unique word types with 1 and 2 occurrence frequency, respectively.

2.3.2 Proposed Methods and Results

Handling Unknown Words

E-commerce product data have many low-frequency words, which leads to the problem of large vocabularies for NMT. Table 2.2 shows the number of unique word types of 1 and 2 frequency. Up to 66% of unique word types occur only once in training corpora. Recently, NMT systems trained on sub-words are widely used to deal with the data sparseness problem.

We examined the impact of sub-words (Sennrich et al., 2016b) on the e-commerce NMT model and compared the results with the output of NMT model trained on word-level. We trained BPE models for each language using Rakuten data, setting BPE merge operations to 16K for each language. Then we tokenized the data from each category using a pre-trained BPE model. For the word-level NMT model, we limited the vocabulary to the top 50K source words and 50K target words by frequency. We set others as an unknown token <*unk*>.

As shown in Table 2.3, the Rakuten BPE-level model displays better performance than the Rakuten word-level model on BLEU score (+6.8 BLEU) and on average RIT score (+0.07 RIT score). For that reason, we decided to train all models on BPE-level[4].

NMT model	BLEU	RIT score
Rakuten all Word	53.77	6.83
Rakuten all BPE	**60.67**	**6.90**

Table 2.3: BLEU and average RIT scores of Rakuten word-level and BPE-level *NMT models* on Rakuten test set.

Granularity

Here, we investigate how granular a translation model should be in order to effectively translate data from each category. Table 2.2 shows that training data size is too small for some categories, especially for Breadmaker, Microwave, and Rice cooker. Rakuten Ichiba contains around 30K categories (Cevahir and Murakami, 2016). It would be nearly impossible to create a translation model for each category; however, in case of an insufficient volume of domain-specific data, adding generic content may help to improve the quality of NMT. Therefore, we concatenate similar in terms of domain datasets to 1) increase the size of training data and 2) to decrease the amount of created NMT models to translate data from each category. To investigate its effect, we trained 4 fine-granular models for different categories with different size of training data from Table 2.2: *Rice cooker all, Shampoo all, Pendant all* and *Tops all*. We also trained medium-granular *Tools all* and *Clothes all* models, that consist of similar domain data, and a coarse-granular *Rakuten all* model using all training data. Then we compared the ability of each model to translate the data from each category.

Table 2.4 shows the BLEU scores of each NMT model on test data of each category. All models demonstrate the best results on their own in-domain data (test data with the same name), except fine-granular *Rice cooker all* model, which was trained on very small data and failed on translating out-of-domain data. On the other hand, medium-granular *Tools all* and *Clothes all* models show the best and

[4] In this section, from now on, we do not include the word “BPE” to the names of models for the sake of simplicity.

the second-best results on in-domain and sub-in-domain (which training data was included in medium-granular training data) datasets. A coarse-granular *Rakuten all* model outperforms *Tools all* on 2 (Breadmaker and Microwave) and *Clothes all* on 3 (Shoes, Skirt and Socks) sub-in-domain datasets. From this point onwards, we experimented with medium-granular and coarse-granular models only.

Test data	*Pendant all*	*Rice cooker all*	*Shampoo all*	*Tops all*	*Tools all*	*Clothes all*	*Rakuten all*
Breadmaker	7.41	5.93	4.73	6.61	32.66	10.04	**45.31**
Microwave	4.15	1.24	3.03	6.68	17.95	12.04	**43.58**
Pendant	**61.72**	0.00	0.63	7.91	59.79	12.83	46.19
Rice cooker	4.91	33.80	3.71	1.76	**47.59**	7.47	30.99
Shampoo	7.43	4.12	**63.76**	4.40	55.58	6.53	28.18
Tools	19.99	10.60	19.11	4.73	**58.73**	8.96	37.21
Shoes	5.08	0.15	0.43	16.45	5.83	62.15	**64.27**
Skirt	5.92	0.00	0.00	31.89	6.63	50.39	**51.59**
Socks	5.84	0.63	0.88	17.23	5.65	54.12	**56.66**
Tops	7.63	0.00	0.30	**62.40**	6.68	54.53	52.47
Clothes	5.99	0.22	0.48	32.62	6.16	**60.34**	56.58
Rakuten	10.14	3.04	4.14	25.12	17.36	44.62	**60.57**

Table 2.4: BLEU scores of *NMT models* on 12 test data. **Bold**: indicates the highest BLEU scores for each test data category.

Data Selection

The quality of the training data plays an important role in training NMT systems. Therefore, selecting high-quality data from a noisy parallel corpus (Imankulova et al., 2017) is considered to be one of the solutions. In this section, we applied data selection from training data and its contribution to the quality of translation for Tools, Clothes, and Rakuten dataset from Table 2.2. For that purpose, we sampled from these training datasets only sentence pairs with RIT score ≥ 3 and trained *Tools sel*, *Clothes sel* and *Rakuten sel* models using the selected data. The sizes of the selected training sets are shown in Table 2.5 (Selected). Development and test sets are the same as in Table 2.2.

The results are shown in Table 2.7 (columns: sel). Compared to the results of NMT models trained on all data (all) from Table 2.4, NMT models trained on the selected data (sel) performed slightly better on an in-domain case and much worse on out-of-domain cases, except for *Rakuten sel* model.

Training data	Selected	Augmented
Tools	132,192	175,687
Clothes	807,307	1,034,577
Rakuten	939,494	1,207,445

Table 2.5: Number of sentence pairs for Selected and Augmented training data.

Additional data	# of sent	orig	*all*	*sel*
Tools	45,155	4.12	4.29	**5.48**
Clothes	242,989	4.56	4.72	**5.17**
Rakuten	288,677	4.68	4.86	**5.36**

Table 2.6: Comparison of average RIT scores. # of sent: number of sentences of additional data; orig: the original English sentences; *all*: output of NMT models trained on all data; *sel*: output of NMT models trained on the selected data.

Data Augmentation

Rakuten Ichiba has a great amount of Japanese monolingual data. Here, we investigate how to effectively use Japanese monolingual data to further improve the quality of NMT models for e-commerce product titles. For that purpose, we used all and sel models to translate in-domain Japanese sentences from Tools, Clothes, and Rakuten training data (Table 2.2) which have RIT score < 3. Table 2.6 shows the size of obtained pseudo-parallel data and the average RIT scores for the original English sentences (orig) and for outputs of all and sel models. In all cases, outputs of sel models are better than that of all and original target sentences. Furthermore, we selected sentences from the pseudo-parallel data to use as additional data to the selected training data from Table 2.5 (Selected). Sentences with the highest RIT score among the outputs of $orig$, all, and sel were kept, and sentences with the RIT score < 3 were eliminated. The sizes of the obtained augmented training data are shown in Table 2.5 (Augmented). Finally, we trained NMT models using the created augmented training data (aug).

BLEU scores of aug models are shown in Table 2.7 (columns: aug). Compared to the results of sel models, aug models are worse for in-domain datasets. However, they demonstrate better results for out-of-domain datasets.

Test data	*Tools*		*Clothes*		*Rakuten*	
	sel	*aug*	*sel*	*aug*	*sel*	*aug*
Tools	**57.32**	36.04	1.95	10.98	36.47	37.24
Clothes	2.60	7.47	**60.79**	57.06	57.59	57.12
Rakuten	9.41	22.85	29.70	47.97	**60.98**	60.93

Table 2.7: BLEU scores of *NMT models* trained on the selected and augmented data.

Test data	orig	*all*	*sel*	*aug*
Tools	6.12	6.17	6.32	**6.73**
Clothes	6.87	6.90	6.89	**7.01**
Rakuten	6.88	6.90	6.97	**7.00**

Table 2.8: Average RIT scores of *NMT models* on in-domain test data.

2.3.3 Discussion

Table 2.8 illustrates the calculated average RIT scores for in-domain data, where we can see that all NMT outputs are better than the original target sentences, which we could not evaluate using BLEU. Also, in contrast to BLEU results (Table 2.7), we can conclude that aug models outperform all other models. We assume that the reasons for such discrepancy between these scores are: 1) NMT models are trying to recreate original data, so they do not correlate with RIT score at some parts; 2) RIT score cares more about how many words, which appear in the equivalent categories on Rakuten.com, are contained in each sentence, while BLEU uses the original data (noisy pair) as the reference; 3) BLEU evaluates from the originally incorrectly translated title, so the “correct” NMT outputs are considered “wrong”.

Table 2.9 shows an example of original and translated sentences from the Rakuten test set. The original target translation of the Japanese word “リボン” is “ribon”, which is the erroneous translation. *Rakuten all Word* model output <*unk*> word translating “リボンシフォンシャツブラウス”. *Rakuten all* and *Rakuten sel* models, which were trained on BPE-level, translated all words, however, output an erroneous translation such as “ribon”. Finally, *Rakuten aug* correctly translated it to “ribbon”.

Model	Model output
Source sentence	楽天大感謝祭！小花リボンシフォンシャツブラウス
Original target sentence	rakuten great thanksgiving ! florets ribon chiffon shirt blouse
Rakuten all Word	rakuten great thanksgiving ! flower <*unk*>
Rakuten all	rakuten great thanksgiving ! florets ribon chiffon shirt blouse
Rakuten sel	rakuten great thanksgiving ! floret ribon chiffon shirt blouse
Rakuten aug	rakuten great thanksgiving ! florets ribbon chiffon shirt blouse

Table 2.9: Example from Rakuten test data translated by *NMT models*.

2.3.4 Summary

In this section, we have explored the possibility of using NMT on e-commerce product titles and demonstrated the effectiveness of using pseudo-parallel corpora created by back-translation and filtering it using RIT score.

2.4 Pseudo-Parallel Corpora by Round-Trip Translation

Since RIT score is a unique evaluation metric that concentrates specifically on English and e-commerce product domain, allowing to evaluate the correctness of the back-translated sentence, such evaluation metrics are not available for most languages and domains. Therefore, in this section, we propose a method to create a pseudo-parallel corpus by translating a monolingual corpus in the target language and filtering it using round-trip translation to address the quality of the parallel corpus. If the target sentence and its round-trip translation are similar, we assume that the synthetic source sentence is appropriate with respect to its monolingual target sentence; moreover, this pair can be included in the filtered pseudo-parallel corpus. The filtration can be iteratively applied using a new upgraded NMT system. Thus, the size of a high-quality pseudo-parallel corpus can be increased. To the best of our knowledge, this study comprises the first attempt to (1) filter a pseudo-parallel corpus using round-trip translation and (2) bootstrap NMT.

Here, we conducted experiments on three different language pairs which have a varying amount of available parallel data. Japanese↔Russian was used as the low-resource language pair, French→Malagasy as medium-resource language pair, and German→English as the high-resource language pair. We demonstrated that the baseline method (Sennrich et al., 2016a) is effective for high-resource language pairs; however, in the case of low-resource language pairs, it is more effective to use a filtered pseudo-parallel corpus as additional training data. With the filtered pseudo-parallel corpus, up to 3.46 BLEU point improvement was achieved in the Russian→Japanese translation, and up to 5.25 BLEU points in the Japanese→Russian translation.

2.4.1 Proposed methods

Filtering

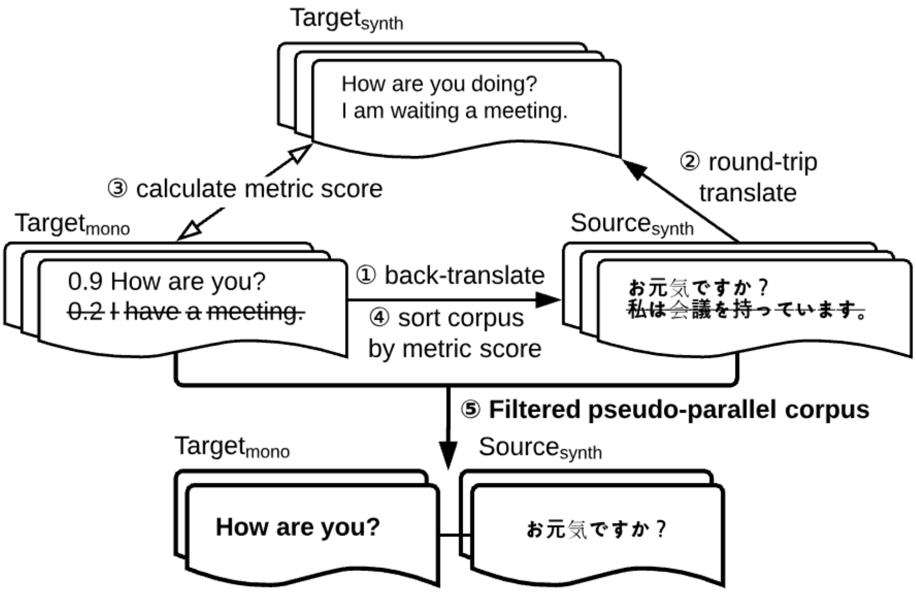


Figure 2.2: Creating and filtering a pseudo-parallel corpus using round-trip translation.

As shown in Figure 2.2, the proposed method includes the following steps:

1. Back-translate monolingual target sentences (Target$_{mono}$) using a *"$Model_b$"* model trained in the target→source direction to produce synthetic source sentences (Source$_{synth}$). Here, an *"Unfiltered"* pseudo-parallel corpus is obtained as additional data without filtration, similar to the approach used in Sennrich et al. (2016a).

2. Round-trip translate the synthetic source sentences using a *“$Model_f$”* model trained in the source→target direction to obtain a synthetic target sentence (Target$_{synth}$).
3. Calculate sentence-level similarity metric scores using the monolingual target sentences as references and the 1-best synthetic target sentences generated via beam-search as candidates.
4. Sort the monolingual target sentences and the corresponding synthetic source sentences in descending order of sentence-level similarity metric scores and filter out sentences with low scores.
5. Use the filtered synthetic source sentences as the source side and the monolingual target sentences as the target side of the pseudo-parallel corpus; this is referred to as a *“Filtered”* pseudo-parallel corpus: it is used as training data in addition to the parallel corpus to train a new *“Filtered”* model.

Bootstrapping

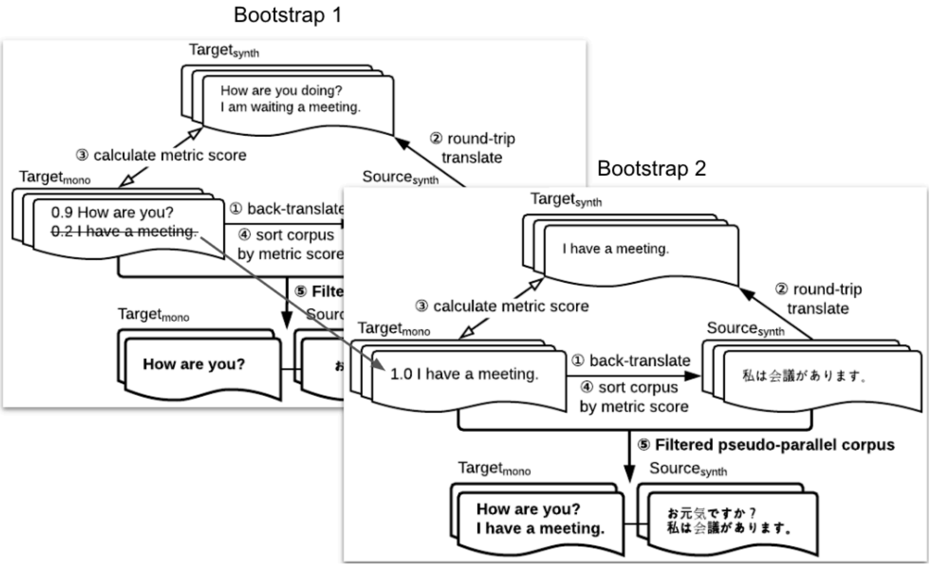


Figure 2.3: Bootstrapping NMT with a pseudo-parallel corpus.

Each bootstrapping iteration involves the following steps (Figure 2.3, Algorithm 1):

Algorithm 1: Bootstrapping NMT using filtered pseudo-parallel corpus

```
Input: Parallel data $src_p \leftrightarrow trg_p$ and monolingual target data $trg_m$, if needed
       $ScoringModel$: word2vec or LM
Output: The best source to target $Model_f$.
$Model_f \leftarrow train(src_p, trg_p)$
$Model_b \leftarrow train(trg_p, src_p)$
$baseScore \leftarrow 0$
$bestScore \leftarrow eval(Model_f)$
$Bootstrap\_iteration \leftarrow 0$
$metric \leftarrow choose(sent\text{-}LM, sent\text{-}BLEU, AAS, MAS)$
while $bestScore > baseScore$ do
    $Bootstrap\_iteration \leftarrow Bootstrap\_iteration + 1$
    $baseScore \leftarrow bestScore$
    $src_s \leftarrow backTranslate(trg_m)$ using $Model_b$
    $trg_s \leftarrow roundTripTranslate(src_s)$ using $Model_f$
    $thresholds \leftarrow filter(metric, src_s, trg_s, trg_m, ScoringModel)$
    for each thr in thresholds do
        $src_n[thr] \leftarrow src_p + src_s$ with $scores \geq thr$
         // parallel + “Filtered” source sentences
        $trg_n[thr] \leftarrow trg_p + trg_m$ with $scores \geq thr$
         // parallel + “Filtered” target sentences
        $Model_f[thr] \leftarrow train(src_n[thr], trg_n[thr])$
         // “Filtered” source to target models
        $bleuScores[thr] \leftarrow eval(Model_f[thr])$
    end
    $thr \leftarrow argmax_{thr}[blueScores]$        // Select thr based on highest BLEU score
    $bestScore \leftarrow bleuScores[thr]$
    $Model_f \leftarrow Model_f[thr]$               // new best “Filtered” source to target model
    $Model_b \leftarrow train(trg_n[thr], src_n[thr])$
     // new best “Filtered” target to source model
    $trg_m \leftarrow trg_m - trg_n[thr]$           // Update with filtered out monolingual data
end
return $Model_f$
```

1. *“Model training”*: Train *“Filtered”* NMT models using a parallel corpus and additional *“Filtered”* pseudo-parallel corpora created by the proposed filtering method.

2. *“Model selection”*: Select the best model on the development set from the

previous iteration and use it instead of the source→target *“$Model_f$”* model from the previous iteration. Additionally, train its target→source *“$Model_b$”* model.

3. *“Bootstrapping”*: Use target sentences from the pseudo-parallel corpus that have been filtered out from training data of the previous best model to create new *“Filtered”* pseudo-parallel corpora to bootstrap the NMT. If there is no improvement over the previous iteration, terminate the bootstrapping process and return to the *“Filtered”* pseudo-parallel corpus and the translation model as output.

4. Repeat steps 1 to 3.

To create a new pseudo-parallel corpus for a new bootstrap iteration, we use those monolingual target sentences that were not included in the *“Filtered”* pseudo-parallel corpus of the previous iteration. Consequently, the already created *“Filtered”* pseudo-parallel corpus from the previous iteration does not change[5] in the next bootstrap iteration. Even if the filtered out monolingual target sentences remain the same, its synthetic source sentences are refreshed at each iteration. Thus, the translation quality of both the *“Unfiltered”* and *“Filtered”* pseudo-parallel corpus is improved via the bootstrapping process until the stopping criterion is met.

Sentence-level similarity metrics for filtering

Three sentence-level similarity metrics are used for filtering: sent-BLEU, AAS, and MAS proposed by Song and Roth (2015), which showed effective results in the Semantic Textual Similarity task (Kajiwara et al., 2017). These metrics require round-trip translation of target monolingual data for the proposed filtration method. Sent-BLEU calculates the similarity of the synthetic and monolingual target sentences based on only surface information, whereas AAS and MAS use distributed representations of the sentences.

The AAS score is the average cosine similarity between vectors of all words in

[5] The attempt to update the entire pseudo-parallel corpus in each bootstrap iteration, instead of using only filtered out monolingual data to create a new pseudo-parallel corpus for a new bootstrap iteration, led to degraded performance.

monolingual and synthetic target sentences:

$$\text{AAS}(y, y') = \frac{1}{|y||y'|} \sum_{i=1}^{|y|} \sum_{j=1}^{|y'|} \cos(\vec{y_i} \vec{y'_j}) \tag{2.1}$$

The MAS score is the cosine similarity between the most similar word from the monolingual target sentence and each word from the synthetic target sentence:

$$\text{MAS}_{asym}(y, y') = \frac{1}{|y|} \sum_{i=1}^{|y|} \underset{j}{max} \cos(\vec{y_i} \vec{y'_j}) \tag{2.2}$$

Note that this similarity is not symmetric. A symmetric similarity can be computed by averaging two similarities:

$$\text{MAS}(y, y') = \frac{1}{2}\text{MAS}_{asym}(y, y') + \frac{1}{2}\text{MAS}_{asym}(y', y) \tag{2.3}$$

Here, $y = (y_1, ..., y_i)$ are word vectors for a monolingual target sentence, and $y' = (y'_1, ..., y'_j)$ are word vectors for a synthetic target sentence.

Sentence-level language model scoring for filtering

We also used the sent-LM metric, which, in contrast to the other three sentence-level similarity metrics, performs filtration by scoring only synthetic source sentences without round-trip translation.

Thresholds for the pseudo-parallel corpus

Accordingly, the translation performance increases as the number of parallel sentences increase (Koehn, 2002). However, for a pseudo-parallel corpus, the translation performance does not necessarily increase with the number of sentences. To determine the effects of the quantity and quality of the pseudo-parallel corpus in NMT, thresholds of metric scores are set with increment steps of 0.1. Thus, pseudo-parallel sentences included as additional data have sentence-level similarity scores greater than or equal to some threshold (e.g., sent-BLEU $\geq$ 0.1,..., sent-BLEU $\geq$ 0.9, ...). Sentences scored and filtered by filtering metrics were used to train the *"Filtered"* models. For example, sentences with filtering metric scores (e.g., sent-BLEU) greater than or equal to 0.1 were used to train the *"sent-BLEU $\geq$ 0.1"* model. Moreover, the NMT system was trained using different thresholds, and their respective performances were compared with development and test sets. Only source→target NMT models with the highest BLEU score on the development set were reported.

2.4.2 Experimental Setup

Toolkits

For the conducted experiments, we used the OpenNMT toolkit[6] (Klein et al., 2017) to train all translation models. For the Russian↔Japanese and French→Malagasy experiments, the following parameters were used: The number of recurrent layers of the encoder and decoder was one, BiLSTM was used with concatenation, the maximum batch size was 32, and the Adadelta optimization method was applied. For the German→English experiments, OpenNMT default settings were used to match the hyper-parameters used for pre-trained German→English models (without back-translation) distributed by OpenNMT.[7] The vocabulary size in all experiments was 50,000.

All French, English, German, and Russian sentences were tokenized and truecased using Moses scripts.[8] For all Japanese sentences, MeCab 0.996 was used with the IPAdic dictionary[9] for word segmentation. For all languages, duplicate sentences and sentences with more than 50 words were eliminated. To compare the translation results, the BLEU scores (Papineni et al., 2002) were recorded. Additionally, Moses's *bootstrap-hypothesis-difference-significance.pl* script was used to perform statistical significance tests on the translations ($p < 0.05$).

The sent-BLEU scores were calculated using the *MTeval-sentence* of the MTeval toolkit.[10] Word2vec (Mikolov et al., 2013a) models were trained using the Gensim library to calculate AAS and MAS metrics. The KenLM Language Model

[6] http://opennmt.net/OpenNMT/
[7] http://opennmt.net/Models/
[8] https://github.com/moses-smt/mosesdecoder/
[9] http://taku910.github.io/mecab
[10] https://github.com/odashi/mteval

Toolkit[11] was used to build a 5-gram language model with Kneser-Ney smoothing. To extract the scores, the filtering metric scores were normalized to be between [0, 1] using a feature-scaling $preprocessing.MinMaxScaler$ method from the scikit-learn library, which transforms features by scaling each feature within the designated min and max range. The LM log probability scores are normalized by dividing the log probability scores by the sentence length.

Dataset

Parallel and target monolingual data The parallel corpora for low-resource Russian↔Japanese[12] and for medium-resource French→Malagasy[13] experiments were downloaded from OPUS.[14] For the medium-resource French-Malagasy language pair, the news domain GlobalVoices corpus was used, which differs from the Tatoeba[15] daily-conversations domain corpus used in the Russian↔Japanese experiments. The GlobalVoices corpus has more available parallel data (107,406 sentence pairs compared with 11,231 available through Tatoeba).

The duplicate Tatoeba parallel corpus was divided for the Russian↔Japanese experiments as follows: 10,231 sentences comprised the training set, 500 sentences the development set, and 500 sentences the test set. Additionally, to perform Japanese→Russian→Japanese round-trip translation for the Russian to Japanese experiment, all available 165,742 Japanese monolingual sentences were sampled from in-domain Tatoeba. All available 75,402 Russian monolingual sentences from in-domain Tatoeba were also sampled for Japanese→Russian translation to facilitate Russian→Japanese→Russian round-trip translation. None of the utilized monolingual data overlapped with the parallel data.

Experiments for the French→Malagasy language pair were conducted with data from the GlobalVoices corpus. Parallel data were divided as follows: 106,406 sentences comprised the training set, 1,000 sentences the development set, and 1,000 sentences the test set. From GlobalVoices, 105,573 Malagasy monolingual sentences were used to create a French→Malagasy pseudo-parallel corpus.

[11] https://kheafield.com/code/kenlm/
[12] http://opus.lingfil.uu.se/Tatoeba.php
[13] http://opus.lingfil.uu.se/GlobalVoices.php
[14] http://opus.lingfil.uu.se/
[15] https://tatoeba.org/eng

For the German→English experiments, pre-trained German↔English models and 4,535,522 parallel sentences provided by OpenNMT were downloaded. The default OpenNMT settings were used to preprocess all data. 4,208,439 German→English sentences from automatically back-translated monolingual data[16] were downloaded; the synthetic German side was translated back to English using the pre-trained German→English model to filter this pseudo-parallel corpus. For the development set, newtest2013 (3,000 sentence pairs) was used, and for the test set, newtest2014 (3,003 sentence pairs). Table 2.10 shows the data statistics.

Corpus	**Russian↔Japanese**	**French→Malagasy**	**German→English**
Train parallel	10,231	106,406	4,535,522
Dev	500	1,000	3,000
Test	500	1,000	3,003
Monolingual target	75,402↔165,742	105,570	4,208,439

Table 2.10: Data statistics.

Data to train word2vec models To train word2vec models for the Russian↔Japanese experiments, the OpenSubtitles2018 corpus[17] was chosen, as its domain is most similar to the Tatoeba domain, with high-resource data. Japanese and Russian monolingual OpenSubtitles2018 corpora were downloaded from OPUS.[18] After tokenizing and removing sentences with less than 2 and more than 100 words, 2,728,314 Japanese monolingual sentences were obtained to train Japanese word2vec. Similarly, Russian monolingual sentences were cleaned, and the same 2,728,314 sentences were sampled to match Japanese monolingual data for fair comparison on low-resource settings. To train the Malagasy word2vec model for the French→Malagasy experiments, data was used from the Leipzig Corpora Collection for Under-resourced Languages[19] (Goldhahn et al., 2016) and from GlobalVoices for all Malagasy monolingual data. The total size of the employed data was 296,440 Malagasy monolingual sentences. For the German→English experiments, the English side of parallel and monolingual data used in the previous Section 2.4.2 was concatenated, resulting in 8,743,962 English monolingual sentences.

[16] http://data.statmt.org/rsennrich/wmt16_backtranslations/de-en/
[17] http://opus.nlpl.eu/OpenSubtitles2018.php/
[18] http://opus.nlpl.eu/index.php
[19] http://curl.corpora.uni-leipzig.de/languages/mlg

Data to train language models To train the language models for the Japanese→Russian and Russian→Japanese experiments, the same Japanese and Russian sentences were used that were used to train the word2vec models. For the French→Malagasy experiments, 2,190,579 French monolingual sentences were used from the Europarl.[20] For the German→English experiments, German monolingual sentences were downloaded from automatically back-translated data[21] and concatenated with the German side of parallel data, resulting in 8,115,406 German monolingual sentences.

Baselines

Sennrich et al. (2016a) obtained additional training data by automatically translating monolingual target sentences into the source language with their *"Parallel"* baseline systems.

The baseline systems used herein were 1) the *"Parallel"* systems trained on a parallel corpus in both directions, which were then used to create a pseudo-parallel corpus, and 2) an *"Unfiltered"* system, which was trained on a concatenated parallel corpus with all pseudo-parallel corpora without any filtration.

Filtering and bootstrapping

The parallel sentence pairs (Section 2.4.2) were used to train the baseline *"Parallel"* models in both directions for all language pairs. Then, these models were used to create a pseudo-parallel corpus by round-trip translation of the target monolingual sentences (Section 2.4.1). A concatenation of parallel and pseudo-parallel sentences was used to train the *"Unfiltered"* models for each bootstrapping iteration. Because of training variance, all *"Unfiltered"* models were trained 10 times with different seeds, as well as one time for each threshold; the scores of the best model on the development set is then reported. For each filtering metric, after filtration and selection of the best model on the development set and its data, the filtered out target monolingual sentences are used for round-trip translation with the chosen model. Since different models are used, *"Unfiltered"* data is created

[20] http://www.statmt.org/wmt14/training-monolingual-europarl-v7/europarl-v7.fr.gz

[21] http://data.statmt.org/rsennrich/wmt16_backtranslations/en-de/

with different quality for each filtering metrics. Here, for each filtering metric, the BLEU scores are compared for the development sets. Iterating was halted if the current score was lower than the previous score. Accordingly, the BLEU scores for development and test sets (Tables 2.11-2.14) for *"Parallel"*, *"Unfiltered"*, and *"Filtered"* models with their data sizes and thresholds (columns "≥") are reported.

2.4.3 Results

In this section, the results on three different language pairs are described: Japanese↔Russian, French →Malagasy, and German→English.

Russian→Japanese

	Metric	Unfiltered				Filtered		
		Size	Dev	Test	≥	Size	Dev	Test
Parallel		10,231	17.47	18.17	-	-	-	-
Bootstrap 1	sent-LM	175,973	18.26	17.87	0.9	175,821	**18.59**	17.27
	sent-BLEU				0.2	72,264	$^{\bullet}_{\circ}$20.93	18.59
	AAS				0.9	34,814	$^{\bullet}_{\circ}$20.00	$^{\bullet}_{\circ}$20.04
	MAS				0.7	48,584	$^{\bullet}_{\circ}$19.95	$^{\bullet}$19.48
Bootstrap 2	sent-LM		17.03	17.27	0.8	175,951	$_{\circ}$18.42	**18.04**
	sent-BLEU		18.74	17.76	0.9	72,291	$^{\bullet}_{\circ}$**21.18**	$^{\bullet}_{\circ}$**19.64**
	AAS		18.74	17.34	1.0	35,032	$^{\bullet}_{\circ}$**21.33**	$^{\bullet}_{\circ}$**20.60**
	MAS		18.55	15.48	0.9	48,593	$^{\bullet}_{\circ}$20.81	$^{\bullet}_{\circ}$20.55
Bootstrap 3	sent-BLEU		18.25	18.15	0.5	73,430	$^{\bullet}_{\circ}$20.67	18.67
	AAS		17.88	16.68	1.0	35,182	$^{\bullet}_{\circ}$19.93	$_{\circ}$19.04
	MAS		18.39	18.61	0.8	55,513	$^{\bullet}_{\circ}$**21.41**	$^{\bullet}_{\circ}$**21.13**
Bootstrap 4	MAS		18.95	19.11	0.9	55,637	$^{\bullet}_{\circ}$21.23	$^{\bullet}$20.18

Table 2.11: Russian→Japanese translation BLEU scores. There is a statistically significant difference for $^{\bullet}$: against the *"Parallel"* system, and for $_{\circ}$: against the *"Unfiltered"* system of that Bootstrap iteration. **Bold** indicates the highest BLEU scores for each filtering metric.

Table 2.11 outlines the results of all bootstrap iterations and filtering metrics for the Russian→Japanese language pair. The results obtained using the *"Unfiltered"* model demonstrate that none of the *"Unfiltered"* models significantly outperformed the *"Parallel"* model.

Contrastingly, *"Filtered"* models significantly outperformed both *"Parallel"* and their *"Unfiltered"* baselines. In most cases, the difference between the BLEU scores of *"Filtered"* and *"Parallel"* on the development set is around +3 points. Generally, these results suggest that using filtered pseudo-parallel data rather than all sentences containing incorrect sentence pairs leads to increased machine translation accuracy for both baselines.

The model trained using data scored by a sent-LM metric stopped improving after the first bootstrap iteration. However, using sentence-level similarity metrics significantly increased the performance over baselines, even when much fewer data were used for the training. The best *"MAS $\geq$ 0.8"* model outperformed the *"Parallel"* and its *"Unfiltered"* model in terms of BLEU scores on the development set by +3.94 and +3.02 points, respectively.

Japanese→Russian

	Metric	Unfiltered				Filtered		
		Size	Dev	Test	$\geq$	Size	Dev	Test
Parallel		10,231	10.13	9.53	-	-	-	-
Bootstrap 1	sent-LM	85,633	$^{\bullet}$14.51	$^{\bullet}$13.96	0.9	84,338	$^{\bullet}$15.30	$^{\bullet}$14.66
	sent-BLEU				0.3	35,811	$^{\bullet}$14.81	13.20
	AAS				0.2	84,934	$^{\bullet}$15.04	13.43
	MAS				0.4	84,762	$^{\bullet}$**15.38**	14.24
Bootstrap 2	sent-LM		$^{\bullet}$14.22	$^{\bullet}$14.00	1.0	84,935	$^{\bullet}_{\circ}$**15.89**	$^{\bullet}$**14.87**
	sent-BLEU		$^{\bullet}$13.69	$^{\bullet}$14.05	0.2	62,519	$^{\bullet}_{\circ}$**15.54**	$^{\bullet}_{\circ}$**15.56**
	AAS		$^{\bullet}$14.60	$^{\bullet}$14.63	0.4	85,101	$^{\bullet}$**15.61**	$^{\bullet}$**15.42**
	MAS		$^{\bullet}$14.24	11.92	0.5	85,152	$^{\bullet}$14.88	$^{\bullet}_{\circ}$**14.51**
Bootstrap 3	sent-LM		$^{\bullet}$14.89	$^{\bullet}$14.20	0.7	85,226	$^{\bullet}$14.89	$^{\bullet}$14.20
	sent-BLEU		$^{\bullet}$14.70	$^{\bullet}$13.08	0.4	62,747	$^{\bullet}$15.35	$^{\bullet}$15.03
	AAS		$^{\bullet}$14.73	$^{\bullet}$14.32	0.8	85,110	$^{\bullet}$15.24	$^{\bullet}$15.07

Table 2.12: Japanese→Russian translation BLEU scores. There is a statistically significant difference, for $^{\bullet}$: against the *"Parallel"* system, and for $_{\circ}$: against the *"Unfiltered"* system of that Bootstrap iteration. **Bold** indicates the highest BLEU scores for each filtering metric.

The effect of the proposed filtering method on Japanese→Russian translations was examined, with the results given in Table 2.12. Most of the *"Unfiltered"* models performed significantly better than the *"Parallel"* model.

However, all *"Filtered"* models still returned better BLEU scores than the *"Parallel"* and *"Unfiltered"* models. The difference between the BLEU scores of the *"Filtered"* and *"Parallel"* models on development and test sets are generally more than +4 points. The *"sent-LM ≥ 1.0"* model in Bootstrap 2 outperformed the *"Parallel"* and its *"Unfiltered"* model in terms of BLEU scores on the development set by +5.76 and +1.67 points, respectively. However, its BLEU score on the test set was low compared to other metrics' models. In Bootstrap 3, The BLEU scores of *"sent-LM ≥ 0.7"* were the same as that of its *"Unfiltered"* model. We assume that the relatively weak models from the previous iteration generated low-quality pseudo-parallel data, which led to weaker *"Filtered"* models that used nearly the same amount of data as the *"Unfiltered"* model.

French→Malagasy

	Metric	Unfiltered				Filtered		
		Size	Dev	Test	≥	Size	Dev	Test
Parallel		106,406	16.78	15.18	-	-	-	-
Bootstrap 1	sent-LM	211,979	17.06	14.90	0.8	211,823	$^{\bullet}$**17.44**	**15.72**
	sent-BLEU				0.3	124,756	$^{\bullet}$17.43	14.39
	AAS				0.9	175,386	$^{\bullet}$17.59	$^{\bullet}$**16.87**
	MAS				0.7	210,851	$^{\bullet}$**17.43**	15.67
Bootstrap 2	sent-LM		17.00	16.06	0.9	211,949	$^{\bullet}$17.35	15.65
	sent-BLEU		16.69	15.32	0.6	125,176	$^{\bullet}_{\circ}$**17.69**	**15.71**
	AAS		17.31	15.35	0.9	189,929	$^{\bullet}$**17.77**	$^{\bullet}$16.26
	MAS		16.80	15.52	0.5	211,840	$^{\bullet}$17.38	**16.06**
Bootstrap 3	sent-BLEU		16.87	16.03	0.9	125,185	17.19	15.64
	AAS		17.07	$^{\bullet}$16.21	0.8	204,817	$^{\bullet}$17.52	15.62

Table 2.13: French→Malagasy translation BLEU scores. There is a statistically significant difference, for $^{\bullet}$: against the *"Parallel"* system, and for $_{\circ}$: against the *"Unfiltered"* system of that Bootstrap iteration. **Bold** indicates the highest BLEU scores for each filtering metric.

The results for French→Malagasy are shown in Table 2.13. Here, only some of the *"Unfiltered"* models slightly outperformed the *"Parallel"* model. Despite the fact that *"Filtered"* models showed higher BLEU scores than both baselines, only some showed significant improvements. On the development set, the best *"AAS ≥ 0.9"* model from Bootstrap 2 yielded better results up to +0.99 BLEU points and +0.46 BLEU points over the *"Parallel"* and *"Unfiltered"* baselines, respectively.

We assume that since French→Malagasy *"Parallel"* models were trained using medium-resource data, they generated a pseudo-parallel corpus with better quality than low-resource language pairs. Even though the filtration method improved baseline models, improvements were not as significant as in low-resource scenarios.

German→English

	Metric	Unfiltered				Filtered		
		Size	Dev	Test	≥	Size	Dev	Test
Parallel		4,535,522	22.33	20.58	-	-	-	-
Bootstrap 1	sent-LM	8,743,961	•**26.35**	•**26.23**	0.9	8,722,498	•25.98	•26.07
	sent-BLEU				0.2	7,345,367	•24.87	•24.13
	AAS				0.3	8,693,417	•26.27	•26.13
	MAS				0.4	8,742,920	•26.14	•25.90

Table 2.14: German→English translation BLEU scores. There is a statistically significant difference, for •: against the *"Parallel"* system, and for ◦: against the *"Unfiltered"* system of that Bootstrap iteration. **Bold** indicates the highest BLEU scores for each filtering metric.

Table 2.14 shows the BLEU scores of German→English experiments. All models that used additional pseudo-parallel corpora achieved significant improvements over the *"Parallel"* baseline. However, none of the *"Filtered"* models outperformed the *"Unfiltered"* baseline on the development and test sets, regardless of filtering metrics.

We assume, in this case, that the *"Parallel"* models were strong enough to generate a large amount of high-quality pseudo-parallel sentences. Thus, using all pseudo-parallel corpus without any filtration improved the *"Parallel"* baseline the most. Additionally, filtering out some noisy pseudo-parallel sentences resulted in weaker models than in the *"Unfiltered"* baseline.

Human evaluation

Additionally, we used the human evaluation of Russian→Japanese translation of the four bootstrap models created using the MAS metric. Two human evaluators were asked to evaluate the translations of 100 source sentences randomly sampled from the test set. Each evaluator chose the best candidate based on adequacy and fluency without knowing which bootstrap system produced the respective translation (ties and "none" were allowed). The final decision was made by voting. The number of times each bootstrap model was selected as the best translation by the evaluators was then calculated.

Table 2.15 shows the results of the human evaluation. The highest number of correct answers for adequacy and fluency was obtained by the *"MAS ≥ 0.8"* model in Bootstrap 3. Thus, bootstrapping had a positive effect on the NMT model.

Iteration	Model	**Adequacy**	**Fluency**
Bootstrap 1	MAS $\geq$ 0.7	21	56
Bootstrap 2	MAS $\geq$ 0.9	16	43
Bootstrap 3	MAS $\geq$ 0.8	**22**	**57**
Bootstrap 4	MAS $\geq$ 0.9	17	48

Table 2.15: Human evaluation of Russian→Japanese translation adequacy and fluency of the bootstrapped models.

2.4.4 Discussion

The results of all experiments showed that, rather than using all additional pseudo-parallel data, the proposed filtering method improved the translation performance in nearly all experiments conducted for low-resource language pairs. The extensive experiments using four different filtering metrics showed that filtering itself significantly impacts low-resource language pairs, as the improvements across different filtering metrics were consistent with the insignificant differences.

Table 2.16 shows examples of Russian→Japanese pseudo-parallel sentences scored by the MAS metric for every bootstrapping iteration. In the first example, the synthetic Russian sentence from Bootstrap 1 was an incorrect translation of its Japanese monolingual sentence. However, by Bootstrap 3, the synthetic Russian and its synthetic Japanese translation improved. By the final Bootstrap 4 iteration, the translation output was sufficient. Finally, these synthetic Russian and Japanese monolingual sentences were used to train the best *"MAS ≥ 0.9"* model from Bootstrap 4. Accordingly, the Japanese→Russian and Russian→Japanese models filtered by the MAS metric used to create the pseudo-parallel corpora improved with each bootstrap iteration. Example 2 shows that the synthetic Russian sentence from Bootstrap 3 was an adequate translation of its Japanese monolingual sentence, whereas its round-trip translation was not. As a result, these pseudo-parallel sentences were filtered out, resulting in a slightly worse synthetic Russian sentence, in terms of adequacy, in Bootstrap 4. Therefore, such noisy sentences are still present in the filtered pseudo-parallel corpus used as additional training data for the best model, which led to the decreased translation quality of the Russian→Japanese NMT model.

Boot	**Synthetic Russian Sentence**	**Synthetic Japanese Sentence**	MAS score
Example 1 - Japanese Monolingual Sentence: ミルクを一杯いかが。**(how about a cup of milk ?)**			
Bootstrap 1	не кофе , пожалуйста . (no coffee , please .)	コ—ヒ— をください。 (coffee , please .)	0.63
Bootstrap 2	не кофе , пожалуйста . (no coffee , please .)	コ—ヒ— を飲んでください。 (please , drink coffee.)	0.54
Bootstrap 3	не могли ли чашку молока ? (couldn’t a cup of milk ?)	牛乳を 1 杯いただけませんか。 (could I have a glass of milk ?)	0.67
Bootstrap 4	не хотите ли чашку молока ? (would you like a cup of milk ?)	ミルクを一杯いかがですか。 (would you like a cup of milk ?)	0.92
Example 2 - Japanese Monolingual Sentence: トムはメアリ— についてあなたに話しましたか。**(did Tom talk to you about Mary ?)**			
Bootstrap 1	Том говорил , что ты говорил ? (Tom said what did you say ?)	トムは何を言っているのだと言ったんだ？ (did you say what Tom is saying?)	0.64
Bootstrap 2	Том говорил с Вами ? (did Tom talk to you ?)	トムはあなたと話しましたか。 (did Tom talk to you ?)	0.76
Bootstrap 3	Том говорил с Вами по Мэри . (Tom spoke to you by Mary .)	トムは一人であなたに会った。 (Tom saw you by yourself .)	0.74
Bootstrap 4	Том говорил с Вами ? (did Tom talk to you ?)	トムはあなたに話しましたか。 (did Tom tell you ?)	0.91

Table 2.16: Examples from the Russian→Japanese pseudo-parallel corpus scored by the MAS metric and their changes on every bootstrapping iteration.

Sennrich et al. (2016a) showed that using a pseudo-parallel corpus as additional data greatly improved the performance over the *“Parallel”* baseline. Furthermore, our experiments demonstrated that for a high-resource language pair (Table 2.14), the proposed filtering method was unable to improve the performance over the *“Unfiltered”* baseline; additionally, it was not as effective as it was for a low-resource language pair. The experiments showed that a better *“Parallel”* system resulted in the creation of a better pseudo-parallel corpus. Thus, the weaker the *“Parallel”* system, the more effective the proposed filtration method.

For some filtration metrics in the experiments of Tables 2.11-2.13, removing only very few of the training data improved BLEU scores over the *“Unfiltered”* baselines. After manual inspection, the removed data were particularly noisy. For example, in Bootstrap 1 of Table 2.12 for the Japanese→Russian experiments, *“sent-LM≥0.9”* removed pairs with a score of 0.55 as follows:

> Synthetic Japanese source sentence: “” 私の父は私の「」は「」”.
>
> Monolingual Russian target sentence: “одна из моих любимых песен”.

We assume that such noisy data greatly affects already poor NMT quality, especially in low-resource scenarios; the NMT system “unlearns” its conditioning on the source context when the training data are noisy. This phenomenon could be addressed as the low-resource case of the work by Sennrich et al. (2016a) with their “Dummy Source Sentences” experiments. It is also known that in low-resource settings, neural networks tend to experience overfitting (Srivastava et al., 2014). Considering the number of parameters, NMT systems tend to overfit on small training data: i.e., they learn both the correct and the noisy information from the given data. The *“Unfiltered”* models are greatly influenced by the training data with respect to the noisy data removed by filtering. However, with filtering, sentences containing such noise can be successfully removed, leading to increased BLEU scores. From this, the utility of filtering is established: the noisy data can be correctly removed from the experiment settings when systems are susceptible to the training data.

The experimental results showed that bootstrapping for multiple iterations improved NMT in terms of the BLEU score. However, the quality ceases to improve after several steps. This could be attributed to the systems used to create new pseudo-parallel corpora, which become weaker at each bootstrap iteration.

The low-resource NMT systems depend not only on the amount of training data but also on the data’s quality. Therefore, even if the number of removed sentences is relatively small, it is more appropriate to define the threshold as an absolute value, rather than a percentage of the data, because the result may change significantly.

2.4.5 Summary

In this section, we showed that we could obtain high-quality pseudo-parallel corpora created by round-trip translation by filtration and bootstrapping for low-resource language pairs. The models trained using the filtered pseudo-parallel corpus as additional data yielded better BLEU scores than the baselines for low-resource language pairs. We also showed that the translation performance could be further improved by bootstrapping, although bootstrapping has its limitations with regard to the BLEU score. These findings suggest that translation accuracy depends on both the size and quality of the training data. The weaker the *"Parallel"* system used in the creation of a pseudo-parallel corpus, the lower the quality of the created pseudo-parallel corpus. In this scenario, the proposed filtration method can be the most useful for obtaining an improved pseudo-parallel corpus.

Further experimental investigations are required to estimate the limitations of the proposed filtration method.

3 | Out-of-Domain Corpora for Low-Resource NMT

3.1 Introduction

In this chapter, we focus on a linguistically distant and thus challenging language pair Japanese↔Russian which has only 12k lines of news domain parallel corpus and hence is extremely resource-poor. Furthermore, the amount of indirect in-domain parallel corpora, i.e., Japanese↔English and Russian↔English, are also small. As we demonstrate in Section 3.4, this severely limits the performance of prominent low-resource techniques, such as multilingual modeling, back-translation, and pivot-based PBSMT. To remedy this, we propose a novel multistage fine-tuning method for NMT that combines multilingual modeling (Johnson et al., 2017) and domain adaptation (Chu et al., 2017).

The main contributions are as follows:

- We have made extensive comparisons with multiple architectures and MT paradigms to show how difficult the problem is. We have also explored the utility of back-translation and show that it is ineffective given the poor performance of base MT systems used to generate pseudo-parallel data. Our systematic exploration shows that multilingualism is extremely useful for in-domain translation with very limited corpora (see Section 3.4). This type of exhaustive exploration has been missing from most existing works.

- Our proposal is to first train a multilingual NMT model on out-of-domain Japanese↔English and Russian↔English data, then fine-tune it on in-domain Japanese↔English and Russian↔English data, and further fine-tune it on Japanese↔Russian data (see Section 3.5). We show that this stage-wise fine-tuning is crucial for high-quality translation. We then show that the improved NMT models lead to pseudo-parallel data of better quality. This data can then be used to improve the performance even further, thereby enabling the generation of better pseudo-parallel data. By iteratively generating pseudo-parallel data and fine-tuning the model on said data, we can achieve the best performance for Japanese↔Russian translation.

- We show that in-domain pivot parallel corpora increase the coverage of Japanese and Russian vocabulary, and it is clarified that the new tokens introduced from in-domain pivot corpora could be translated successfully (see Section 3.6).

To the best of our knowledge, we are the first to perform such an extensive evaluation of an extremely low-resource MT problem and propose a novel multilingual multistage fine-tuning approach involving multilingual modeling and domain adaptation to address it.

3.2 Japanese–Russian Setting

In this chapter, we deal with Japanese↔Russian news translation. This language pair is very challenging because the languages involved have completely different writing systems, phonology, morphology, grammar, and syntax. Among various domains, we experimented with translations in the news domain, considering the importance of sharing news between different language speakers. Moreover, the news domain is one of the most challenging tasks due to the large presence of out-of-vocabulary (OOV) tokens and long sentences.[1] To establish and evaluate existing methods, we also involved English as the third language. As direct parallel corpora are scarce, involving a language such as English for pivoting is quite common (Utiyama and Isahara, 2007).

There has been no clean held-out parallel data for Japanese↔Russian and Japanese↔English news translation. Therefore, we manually compiled development and test sets using News Commentary data[2] as a source.

Specifically, we carried out the following procedure.

1. Given Japanese↔Russian and Japanese↔English data share many Japanese lines; we thus first compiled tri-text data.

2. From each line, corresponding parts across languages were manually identified, and unaligned parts were split off into a new line. Note that we have never merged two or more lines. As a result, we obtained 1,654 lines of data

[1] News domain translation is also the most competitive task in WMT, indicating its importance.
[2] http://opus.nlpl.eu/News-Commentary-v11.php

comprising trilingual, bilingual, and monolingual segments (mainly sentences) as summarized in Table 3.1. Among created 1,086 Japanese↔Russian sentence pairs and 1189 English↔Japanese sentence pairs, 913 pairs were trilingual.

3. We randomly chose 600 trilingual sentences to create a test set for comparability. Then, we concatenated the rest of them and bilingual sentences to form development sets.

Ru	Ja	En	#sent.	Usage	
				test	development
✓	✓	✓	913	600	313
✓	✓		173	-	173
	✓	✓	276	-	276
✓		✓	0	-	-
✓			4	-	-
	✓		287	-	-
		✓	1	-	-
Total			1,654	-	-

Table 3.1: Manually aligned News Commentary data.

Our manually aligned development and test sets are publicly available.[3]

3.3 Related Work

Koehn and Knowles (2017) showed that NMT is unable to handle low-resource language pairs as opposed to PBSMT. Transfer learning approaches (Firat et al., 2016; Zoph et al., 2016; Kocmi and Bojar, 2018) work well when a large helping parallel corpus is available. This restricts one of the sources or the target languages to be English, which, in our case, is not possible. Approaches involving bi-directional NMT modeling is shown to drastically improve low-resource translation (Niu et al., 2018). However, like most other, this work focuses on translation from and into English.

Remaining options include (a) unsupervised MT (Artetxe et al., 2018; Lample et al., 2018b; Marie and Fujita, 2018), (b) parallel sentence mining from non-parallel or comparable corpora (Utiyama and Isahara, 2003; Tillmann and Xu,

[3] https://github.com/aizhanti/JaRuNC

2009), (c) generating pseudo-parallel data (Sennrich et al., 2016a), and (d) MT based on pivot languages (Utiyama and Isahara, 2007). The linguistic distance between Japanese and Russian makes it extremely difficult to learn bilingual knowledge, such as bilingual lexicons and bilingual word embeddings. Unsupervised MT is thus not promising yet, due to its heavy reliance on accurate bilingual word embeddings. Neither does parallel sentence mining, due to the difficulty of obtaining accurate bilingual lexicons. Pseudo-parallel data can be used to augment existing parallel corpora for training, and previous work has reported that such data generated by so-called back-translation can substantially improve the quality of NMT. However, this approach requires base MT systems that can generate somewhat accurate translations. It is thus infeasible in our scenario because we can obtain only a weak system, which is the consequence of an extremely low-resource situation. MT-based on pivot languages requires large in-domain parallel corpora involving the pivot languages. This technique is thus infeasible because the in-domain parallel corpora for Japanese↔English and Russian↔English pairs are also extremely limited, whereas there are large parallel corpora in other domains. Section 3.4 empirically confirms the limit of these existing approaches.

Fortunately, there are two useful transfer learning solutions using NMT: (e) multilingual modeling to incorporate multiple language pairs into a single model (Johnson et al., 2017) and (f) domain adaptation to incorporate out-of-domain data (Chu et al., 2017). In this section, we explore a novel method involving step-wise fine-tuning to combine these two methods. By improving the translation quality in this way, we can also increase the likelihood of pseudo-parallel data being useful to further improve translation quality.

3.4 Limit of Using only In-domain Data

This section is about the translation quality that we can achieve using existing methods and in-domain parallel and monolingual data. We then use the strongest model to conduct experiments on generating and utilizing back-translated pseudo-parallel data for augmenting NMT. Our intention is to empirically identify the most effective practices as well as recognize the limitations of relying only on in-domain parallel corpora.

3.4.1 Data

To train MT systems among the three languages, i.e., Japanese, Russian, and English, we used all the parallel data provided by Global Voices,[4] more specifically those available at OPUS.[5] Table 3.2 summarizes the size of train/development/test splits used in our experiments. The number of parallel sentences for Japanese↔Russian is 12k, for Japanese↔English is 47k, and for Russian↔English is 82k. Note that the three corpora are not mutually exclusive: 9k out of 12k sentences in the Japanese↔Russian corpus were also included in the other two parallel corpora, associated with identical English translations. This puts a limit on the positive impact that the helping corpora can have on the translation quality.

Lang.pair	Partition	#sent.	#tokens	#types
Ja↔Ru	train	12,356	341k / 229k	22k / 42k
	development	486	16k / 11k	2.9k / 4.3k
	test	600	22k / 15k	3.5k / 5.6k
Ja↔En	train	47,082	1.27M / 1.01M	48k / 55k
	development	589	21k / 16k	3.5k / 3.8k
	test	600	22k / 17k	3.5k / 3.8k
Ru↔En	train	82,072	1.61M / 1.83M	144k / 74k
	development	313	7.8k / 8.4k	3.2k / 2.3k
	test	600	15k / 17k	5.6k / 3.8k

Table 3.2: Statistics on our in-domain parallel data.

Even when one focuses on low-resource language pairs, we often have access to larger quantities of in-domain monolingual data of each language. Such monolingual data are useful to improve the quality of MT, for example, as the source of pseudo-parallel data for augmenting training data for NMT (Sennrich et al., 2016a) and as the training data for large and smoothed language models for PBSMT (Koehn and Knowles, 2017). Table 3.3 summarizes the statistics on our monolingual corpora for several domains, including the news domain. Note that we removed from the Global Voices monolingual corpora those sentences that are already present in the parallel corpus.

[4] https://globalvoices.org/

[5] http://opus.nlpl.eu/GlobalVoices-v2015.php

Corpus	Ja	Ru	En
Global Voices[5]	26k	24k	842k
Wikinews[6]	37k	243k	-
News Crawl[7]	-	72M	194M
Yomiuri (2007–2011)[8]	19M	-	-
IWSLT[9]	411k	64k	66k
Tatoeba[10]	5k	58k	208k

Table 3.3: Number of lines in our monolingual data. Whereas the first four are from the news corpora (in-domain), the last two, i.e., “IWSLT” and “Tatoeba,” are from other domains.

We tokenized English and Russian sentences using *tokenizer.perl* of `Moses` (Koehn et al., 2007).[11] To tokenize Japanese sentences, we used `MeCab`[12] with the IPA dictionary. After tokenization, we eliminated duplicated sentence pairs and sentences with more than 100 words for all the languages.

3.4.2 MT Methods Examined

We began with evaluating standard MT paradigms, i.e., PBSMT (Koehn et al., 2007) and NMT (Sutskever et al., 2014). As for PBSMT, we also examined two advanced methods: pivot-based translation relying on a helping language (Utiyama and Isahara, 2007) and induction of phrase tables from monolingual data (Marie and Fujita, 2018) (Figure 3.1).

As for NMT, we compared two types of encoder-decoder architectures: attentional RNN-based model (RNMT) (Bahdanau et al., 2014) and the Transformer model (Vaswani et al., 2017). In addition to standard uni-directional modeling, to cope with the low-resource problem, we examined two multi-directional models: bi-directional model (Niu et al., 2018) and multi-to-multi (M2M) model (Johnson et al., 2017).

After identifying the best model, we also examined the usefulness of a data aug-

[6] https://dumps.wikimedia.org/backup-index.html (20180501)
[7] http://www.statmt.org/wmt18/translation-task.html
[8] https://www.yomiuri.co.jp/database/glossary/
[9] http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/
[10] http://opus.nlpl.eu/Tatoeba-v2.php
[11] https://github.com/moses-smt/mosesdecoder
[12] http://taku910.github.io/mecab, version 0.996.

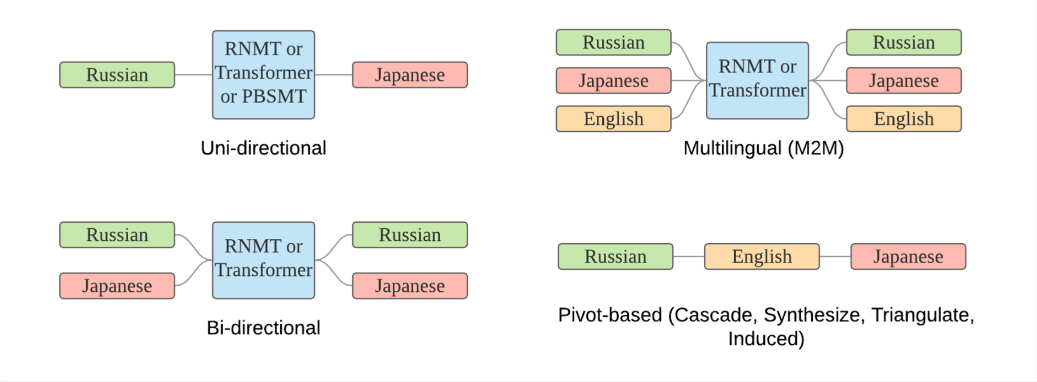


Figure 3.1: Examined models.

mentation method based on back-translation (Sennrich et al., 2016a).

PBSMT Systems

First, we built a PBSMT system for each of the six translation directions. We obtained phrase tables from parallel corpus using `SyMGIZA++`[13] with the `grow-diag-final` heuristics for word alignment, and `Moses` for phrase pair extraction. Then, we trained a bi-directional MSD (monotone, swap, and discontinuous) lexicalized reordering model. We also trained three 5-gram language models, using `KenLM`[14] on the following monolingual data: (1) the target side of the parallel data, (2) the concatenation of (1) and the monolingual data from Global Voices, and (3) the concatenation of (1) and all monolingual data in the news domain in Table 3.3.

Subsequently, using English as the pivot language, we examined the following three types of pivot-based PBSMT systems (Utiyama and Isahara, 2007; Cohn and Lapata, 2007) for each of Japanese→Russian and Russian→Japanese.

Cascade: 2-step decoding using the source-to-English and English-to-target systems.

Synthesize: Obtain a new phrase table from synthetic parallel data generated by translating the English side of the target–English training parallel data to the source language with the English-to-source system.

[13] https://github.com/emjotde/symgiza-pp
[14] https://github.com/kpu/kenlm

Triangulate: Compile a new phrase table combining those for the source-to-English and English-to-target systems.

Among these three, triangulation is the most computationally expensive method. Although we had filtered the component phrase tables using the statistical significance pruning method (Johnson et al., 2007), triangulation can generate an enormous number of phrase pairs. To reduce the computational cost during decoding and the negative effects of potentially noisy phrase pairs, we retained for each source phrase s only the k-best translations t according to the forward translation probability $\phi(t|s)$ calculated from the conditional probabilities in the component models as defined in Utiyama and Isahara (2007). For each of the retained phrase pairs, we also calculated the backward translation probability, $\phi(s|t)$, and lexical translation probabilities, $\phi_{\mathit{lex}}(t|s)$ and $\phi_{\mathit{lex}}(s|t)$, in the same manner as $\phi(t|s)$.

We also investigated the utility of recent advances in unsupervised MT. Even though we began with a publicly available implementation of unsupervised PB-SMT (Lample et al., 2018b),[15] it crashed due to unknown reasons. We, therefore, followed another method described in Marie and Fujita (2018). Instead of short n-grams (Artetxe et al., 2018; Lample et al., 2018b), we collected a set of phrases in Japanese and Russian from respective monolingual data using the `word2phrase` algorithm (Mikolov et al., 2013b),[16] as in Marie and Fujita (2018). To reduce the complexity, we used randomly selected 10M monolingual sentences, and 300k most frequent phrases made of words among the 300k most frequent words. For each source phrase s, we selected 300-best target phrases t according to the translation probability as in Lample et al. (2018b): $p(t|s) = \frac{\exp(\beta \cos(\mathit{emb}(t), \mathit{emb}(s))}{\sum_{t'} \exp(\beta \cos(\mathit{emb}(t'), \mathit{emb}(s))}$, where $\mathit{emb}(\cdot)$ stands for a bilingual embedding of a given phrase, obtained through averaging bilingual embeddings of constituent words learned from the two monolingual data using `fastText`[17] and `vecmap`.[18] For each of the retained phrase pair, $p(s|t)$ was computed analogously. We also computed lexical translation probabilities relying on those learned from the given small parallel corpus.

Up to four phrase tables were jointly exploited by the multiple decoding path abil-

[15] https://github.com/facebookresearch/UnsupervisedMT

[16] https://code.google.com/archive/p/word2vec/

[17] https://fasttext.cc/

[18] https://github.com/artetxem/vecmap

ity of Moses. Weights for the features were tuned using KB-MIRA (Cherry and Foster, 2012) on the development set; we took the best weights after 15 iterations. Two hyper-parameters, namely, k for the number of pivot-based phrase pairs per source phrase and d for distortion limit, were determined by a grid search on $k \in \{10, 20, 40, 60, 80, 100\}$ and $d \in \{8, 10, 12, 14, 16, 18, 20\}$. In contrast, we used predetermined hyper-parameters for phrase table induction from monolingual data, following the convention: 200 for the dimension of word and phrase embeddings and $\beta = 30$.

NMT Systems

We used the open-source implementation of the RNMT and the Transformer models in tensor2tensor.[19] A uni-directional model for each of the six translation directions was trained on the corresponding parallel corpus. Bi-directional and M2M models were implemented by adding an artificial token that specifies the target language to the beginning of each source sentence and shuffling the entire training data (Johnson et al., 2017).

ID	System	Parallel data			Total size of	Vocabulary
		Ja↔Ru	Ja↔En	Ru↔En	training data	size
(a1), (b1)	Ja→Ru or Ru→Ja	12k	-	-	12k	16k
	Ja→En or En→Ja	-	47k	-	47k	16k
	Ru→En or En→Ru	-	-	82k	82k	16k
(a2), (b2)	Ja→Ru and Ru→Ja	12k	-	-	24k	16k
	Ja→En and En→Ja	-	47k	-	94k	16k
	Ru→En and En→Ru	-	-	82k	164k	16k
(a3), (b3)	M2M systems	12k→82k	47k→82k	82k	492k	32k

Table 3.4: Configuration of uni-, bi-directional, and M2M NMT baseline systems. Arrows in "Parallel data" columns indicate the over-sampling of the parallel data to match the size of the largest parallel data.

Table 3.4 contains some specific hyper-parameters[20] for our baseline NMT models. The hyper-parameters not mentioned in this table used the default values in tensor2tensor. For M2M systems, we over-sampled Japanese→Russian and Japanese→English training data so that their sizes match the largest Russian→English

[19] https://github.com/tensorflow/tensor2tensor, version 1.6.6.

[20] We compared two mini-batch sizes, 1024 and 6144 tokens, and found that 6144 and 1024 worked better for RNMT and Transformer, respectively.

data. To reduce the number of unknown words, we used tensor2tensor's internal sub-word segmentation mechanism. Since we work in a low-resource setting, we used shared sub-word vocabularies of size 16k for the uni- and bi-directional models and 32k for the M2M models. The number of training iterations was determined by early-stopping: we evaluated our models on the development set every 1,000 updates and stopped training if the BLEU score for the development set was not improved for 10,000 updates (10 check-points). Note that the development set was created by concatenating those for the individual translation directions without any over-sampling.

Having trained the models, we averaged the last 10 check-points and decoded the test sets with a beam size of 4 and a length penalty (Wu et al., 2016) which was tuned by a linear search on the BLEU score for the development set.

Similarly to PBSMT, we also evaluated "Cascade" and "Synthesize" methods with uni-directional NMT models.

3.4.3 Results

We evaluated the MT models using case-sensitive and tokenized BLEU (Papineni et al., 2002) on test sets, using Moses's *multi-bleu.perl*. Statistical significance ($p < 0.05$) on the difference of BLEU scores was tested by Moses's *bootstrap-hypothesis-difference-significance.pl*.

ID	System	Ja→Ru	Ru→Ja	Ja→En	En→Ja	Ru→En	En→Ru
(a1)	Uni-directional RNMT	0.58	1.86	2.41	7.83	18.42	13.64
(a2)	Bi-directional RNMT	0.65	1.61	6.18	8.81	19.60	15.11
(a3)	M2M RNMT	1.51	4.29	5.15	7.55	14.24	10.86
(b1)	Uni-directional Transformer	0.70	1.96	4.36	7.97	20.70	16.24
(b2)	Bi-directional Transformer	0.19	0.87	6.48	10.63	22.25	16.03
(b3)	M2M Transformer	**3.72**	**8.35**	**10.24**	**12.43**	22.10	**16.92**
(c1)	Uni-directional supervised PBSMT	2.02	4.45	8.19	10.27	**22.37**	16.52

Table 3.5: BLEU scores of baseline systems. **Bold** indicates the best BLEU score for each translation direction.

Tables 3.5 and 3.6 show BLEU scores of all the models, except the NMT systems augmented with back-translations. Whereas some models achieved reasonable

BLEU scores for Japanese↔English and Russian↔English translation, all the results for Japanese↔Russian, which is our main concern, were abysmal.

Among the NMT models, Transformer models (b∗) were proven to be better than RNMT models (a∗). RNMT models could not even outperform the uni-directional PBSMT models (c1). M2M models (a3) and (b3) outperformed their corresponding uni- and bi-directional models in most cases. It is worth noting that in this extremely low-resource scenario, BLEU scores of the M2M RNMT model for the largest language pair, i.e., Russian↔English, were lower than those of the uni- and bi-directional RNMT models as in Johnson et al. (2017). In contrast, with the M2M Transformer model, Russian↔English also benefited from multilingualism.

System	Ja→Ru	Ru→Ja
PBSMT: Cascade	3.65	7.62
PBSMT: Synthesize	3.37	6.72
PBSMT: Synthesize / Gold	2.94	6.95
PBSMT: Synthesize + Gold	3.07	6.62
PBSMT: Triangulate	**3.75**	7.02
PBSMT: Triangulate / Gold	**3.93**	7.02
PBSMT: Synthesize / Triangulate / Gold	**4.02**	7.07
PBSMT: Induced	0.37	0.65
PBSMT: Induced / Synthesize / Triangulate / Gold	2.85	6.86
RNMT: Cascade	1.19	6.73
RNMT: Synthesize	1.82	3.02
RNMT: Synthesize + Gold	1.62	3.24
Transformer NMT: Cascade	2.41	6.84
Transformer NMT: Synthesize	1.78	5.43
Transformer NMT: Synthesize + Gold	2.13	5.06
(c1) Uni-directional supervised PBSMT in Table 3.5	2.02	4.45

Table 3.6: BLEU scores of pivot-based systems. “Gold” refers to the phrase table trained on the parallel data. **Bold** indicates the BLEU score higher than the best one in Table 3.5. “/” indicates the use of separately trained multiple phrase tables, whereas so does “+” training on the mixture of parallel data.

Standard PBSMT models (c1) achieved higher BLEU scores than uni-directional NMT models (a1) and (b1), as reported by Koehn and Knowles (2017), whereas they underperform the M2M Transformer NMT model (b3). As shown in Table 3.6, pivot-based PBSMT systems always achieved higher BLEU scores than (c1). The best model with three phrase tables, labeled “Synthesize / Triangulate / Gold,” brought visible BLEU gains with substantial reduction of OOV tokens

(3047→1180 for Japanese→Russian, 4463→1812 for Russian→Japanese). However, further extension with phrase tables induced from monolingual data did not push the limit, despite their high coverage; only 336 and 677 OOV tokens were left for the two translation directions, respectively. This is due to the poor quality of the bilingual word embeddings used to extract the phrase table, as envisaged in Section 3.3.

None of the pivot-based approaches with uni-directional NMT models could even remotely rival the M2M Transformer NMT model (b3).

3.4.4 Augmentation with Back-translation

Given that the M2M Transformer NMT model (b3) achieved the best results for most of the translation directions and competitive results for the rest, we further explored it through back-translation.

We examined the utility of pseudo-parallel data for all the six translation directions, unlike the work of Lakew et al. (2017) and Lakew et al. (2018), which concentrate only on the zero-shot language pair, and the work of Niu et al. (2018), which compares only uni- or bi-directional models. We investigated whether each translation direction in M2M models will benefit from pseudo-parallel data, and if so, what kind of improvement takes place.

First, we selected sentences to be back-translated from in-domain monolingual data (Table 3.3), relying on the score proposed by Moore and Lewis (2010) via the following procedure.

1. For each language, train two 4-gram language models, using KenLM: an in-domain one on all the Global Voices data, i.e., both parallel and monolingual data, and a general-domain one on the concatenation of Global Voices, IWSLT, and Tatoeba data.

2. For each language, discard sentences containing OOVs according to the in-domain language model.

3. For each translation direction, select the T-best monolingual sentences in the news domain, according to the difference between cross-entropy scores given by the in-domain and general-domain language models.

Whereas Niu et al. (2018) exploited monolingual data much larger than parallel data, we maintained a 1:1 ratio between them (Johnson et al., 2017), setting T to the number of lines of parallel data of given language pair.

Selected monolingual sentences were then translated using the M2M Transformer NMT model (b3) to compose pseudo-parallel data. Then, the pseudo-parallel data were enlarged by over-sampling as summarized in Table 3.7. Finally, new NMT models were trained on the concatenation of the original parallel and pseudo-parallel data from scratch in the same manner as the previous NMT models with the same hyper-parameters.

ID	System		Parallel data			Total size of
		Pseudo	Ja↔Ru	Ja↔En	Ru↔En	training data
#1–#10	Ja∗→Ru and/or Ru∗→Ja	12k→82k	12k→82k	47k→82k×2	82k×2	984k
	Ja∗→En and/or En∗→Ja	47k→82k	12k→82k×2	47k→82k	82k×2	984k
	Ru∗→En and/or En∗→Ru	82k	12k→82k×2	47k→82k×2	82k	984k
	All	All of the above	12k→82k	47k→82k	82k	984k

Table 3.7: Over-sampling criteria for pseudo-parallel data generated by back-translation.

ID	Pseudo-parallel data involved						BLEU score					
	Ja∗→Ru	Ru∗→Ja	Ja∗→En	En∗→Ja	Ru∗→En	En∗→Ru	Ja→Ru	Ru→Ja	Ja→En	En→Ja	Ru→En	En→Ru
(b3)	-	-	-	-	-	-	3.72	8.35	10.24	12.43	22.10	16.92
#1	✓	-	-	-	-	-	**•4.59**	**8.63**	**10.64**	**12.94**	**22.21**	**17.30**
#2	-	✓	-	-	-	-	**3.74**	**•8.85**	10.13	**13.05**	**22.48**	**17.20**
#3	✓	✓	-	-	-	-	**•4.56**	**•9.09**	**10.57**	**•13.23**	**22.48**	**•17.89**
#4	-	-	✓	-	-	-	3.71	8.05	**•11.00**	**12.66**	**22.17**	16.76
#5	-	-	-	✓	-	-	3.62	8.10	9.92	**•14.06**	21.66	16.68
#6	-	-	✓	✓	-	-	3.61	7.94	**•11.51**	**•14.38**	**22.22**	16.80
#7	-	-	-	-	✓	-	**3.80**	**8.37**	**10.67**	**13.00**	**22.51**	**•17.73**
#8	-	-	-	-	-	✓	**3.77**	8.04	**10.52**	12.43	**•22.85**	**17.13**
#9	-	-	-	-	✓	✓	3.37	8.03	10.19	**12.79**	**22.77**	**17.26**
#10	✓	✓	✓	✓	✓	✓	**•4.43**	**•9.38**	**•12.06**	**•14.43**	**•23.09**	**17.30**

Table 3.8: BLEU scores of M2M Transformer NMT systems trained on the mixture of given parallel corpus and pseudo-parallel data generated by back-translation using (b3). Six "X∗→Y" columns show whether the pseudo-parallel data for each translation direction is involved. **Bold** indicates the scores higher than (b3) and "•" indicates statistical significance of the improvement.

Table 3.8 shows the BLEU scores achieved by several reasonable combinations of six-way pseudo-parallel data. We observed that the use of all six-way pseudo-parallel data (#10) significantly improved the base model for all the translation directions, except English→Russian. A translation direction often benefited when the pseudo-parallel data for that specific direction was used.

3.4.5 Summary

We have evaluated an extensive variation of MT models[21] that rely only on in-domain parallel and monolingual data. However, the resulting BLEU scores for Japanese→Russian and Russian→Japanese tasks do not exceed 10 BLEU points, implying the inherent limitation of the in-domain data as well as the difficulty of these translation directions.

3.5 Exploiting Large Out-of-Domain Data Involving a Helping Language

The limitation of relying only on in-domain data demonstrated in Section 3.4 motivates us to explore other types of parallel data. Therefore, we considered effective ways to exploit out-of-domain data.

Domain \ language pair	Direct	One-side shared
in-domain	A, ✓	B, ✓
out-of-domain	C, ×	D, ✓

Table 3.9: Classification of parallel data. "Direct" column indicates the same language pair of interest, here, Japanese↔Russian. "One-side shared" column indicates helping language pairs, such as Japanese↔English and Russian↔English.

According to language pair and domain, parallel data can be classified into four categories in Table 3.9. Among all the categories, out-of-domain data for the language pair of interest have been exploited in the domain adaptation scenarios (C→A) (Chu et al., 2017). However, for Japanese↔Russian, no out-of-domain data is available. To exploit out-of-domain parallel data for Japanese↔English

[21] Other conceivable options include transfer learning using parallel data between English and one of Japanese and Russian as either source or target language, such as pre-training an English→Russian model and fine-tuning it for Japanese→Russian. Our M2M models conceptually subsume them, even though they do not explicitly divide the two steps during training. On the other hand, our method proposed in Section 3.5 explicitly conduct transfer learning for domain adaptation followed by additional transfer learning across different languages.

and Russian↔English pairs instead, we propose a multistage fine-tuning method, which combines two types of transfer learning, i.e., domain adaptation for Japanese↔English and Russian↔English (D→B) and multilingual transfer (B→A), relying on the M2M model examined in Section 3.4 (Figure 3.2). We also examined the utility of fine-tuning for iteratively generating and using pseudo-parallel data.

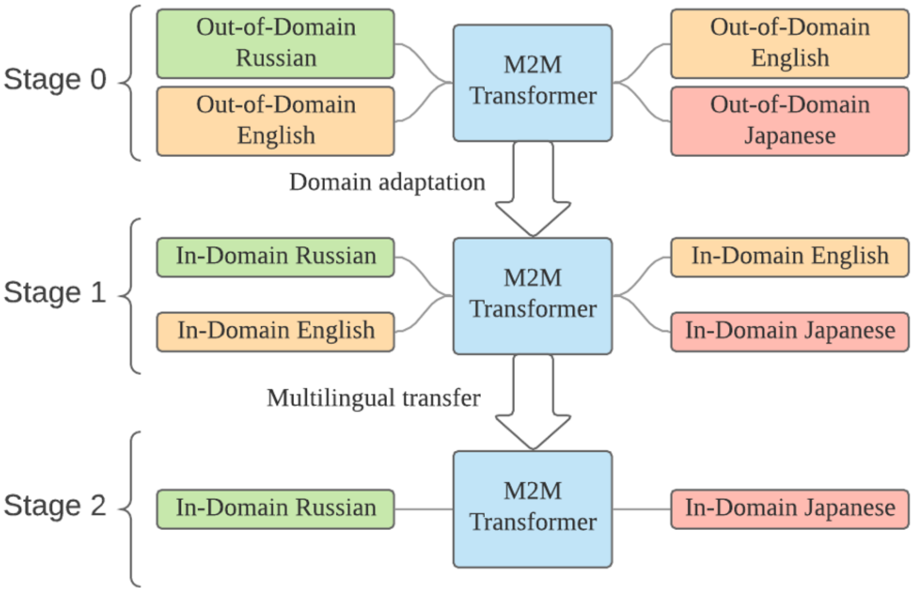


Figure 3.2: Overall illustration of multi-stage fine-tuning.

3.5.1 Multistage Fine-tuning

Simply using NMT systems trained on out-of-domain data for in-domain translation is known to perform badly (Haddow and Koehn, 2012; Koehn and Knowles, 2017). In order to effectively use large-scale out-of-domain data for our extremely low-resource task, we propose to perform domain adaptation through either (a) conventional fine-tuning, where an NMT system trained on out-of-domain data is fine-tuned only on in-domain data, or (b) mixed fine-tuning (Chu et al., 2017), where pre-trained out-of-domain NMT system is fine-tuned using a mixture of in-domain and out-of-domain data. The same options are available for transferring from Japanese↔English and Russian↔English to Japanese↔Russian.

We inevitably involve two types of transfer learning, i.e., domain adaptation for Japanese↔English and Russian↔English and multilingual transfer for Japanese↔Russian pair. Among several conceivable options for managing these two problems, we examined the following multistage fine-tuning (Figure 3.3).

Stage 0. Out-of-domain pre-training: Pre-train a multilingual model only on the Japanese↔English and Russian↔English out-of-domain parallel data (I), where the vocabulary of the model is determined on the basis of the in-domain parallel data in the same manner as the M2M NMT models examined in Section 3.4.

Stage 1. Fine-tuning for domain adaptation: Fine-tune the pre-trained model (I) on the in-domain Japanese↔English and Russian↔English parallel data (fine-tuning, II) or on the mixture of in-domain and out-of-domain Japanese↔English and Russian↔English parallel data (mixed fine-tuning, III).

Stage 2. Fine-tuning for Japanese↔Russian pair: Further fine-tune the models (each of II and III) for Japanese↔Russian on in-domain parallel data for this language pair only (fine-tuning, IV and VI) or on all the in-domain parallel data (mixed fine-tuning, V and VII).

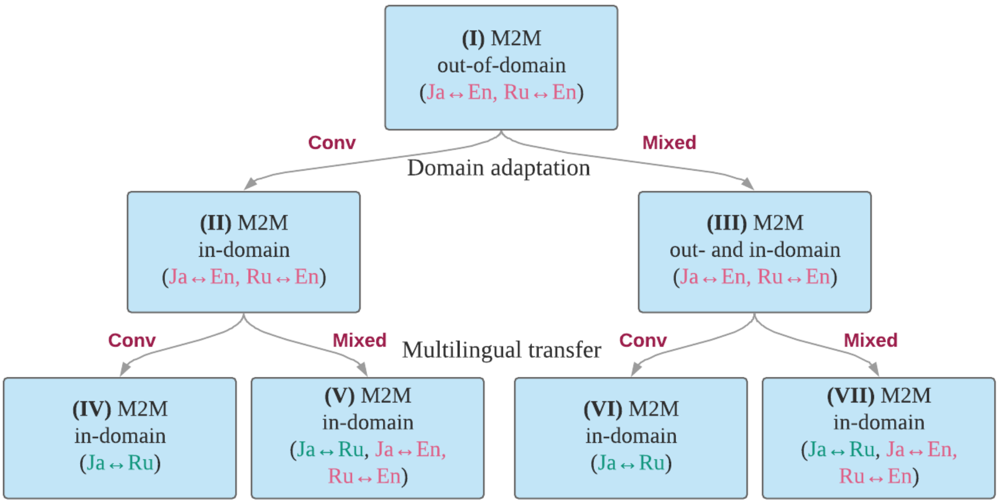


Figure 3.3: Variants of fine-tuning methods.

We chose this way due to the following two reasons. First, we need to take a balance between several different parallel corpora sizes. The other reason is a division of labor; we assume that solving each sub-problem one by one should enable a gradual shift of parameters.

3.5.2 Data Selection

As an additional large-scale out-of-domain parallel data for Japanese↔English, we used the first 1.5M sentences from the Asian Scientific Paper Excerpt Corpus (ASPEC) (Nakazawa et al., 2016).[22] As for Russian↔English, we used the UN and Yandex corpora released for the WMT 2018 News Translation Task.[23] We retained Russian↔English sentence pairs that contain at least one OOV token in both sides, according to the in-domain language model trained in Section 3.4.4. Table 3.10 summarizes the statistics on the remaining out-of-domain parallel data.

Lang.pair	Corpus	#sent.	#tokens	#types
Ja↔En	ASPEC	1,500,000	42.3M / 34.6M	234k / 1.02M
Ru↔En	UN	2,647,243	90.5M / 92.8M	757k / 593k
	Yandex	320,325	8.51M / 9.26M	617k / 407k

Table 3.10: Statistics on our out-of-domain parallel data.

3.5.3 Results

Table 3.11 shows the results of our multistage fine-tuning, where the IDs of each row refer to those described in Section 3.5.1. First of all, the final models of our multistage fine-tuning, i.e., V and VII, achieved significantly higher BLEU scores than (b3) in Table 3.5, a weak baseline without using any monolingual data, and #10 in Table 3.8, a strong baseline established with monolingual data.

The performance of the initial model (I) depends on the language pair. For Japanese↔Russian pair, it cannot achieve a minimum level of quality since the model has never seen parallel data for this pair. The performance on Japanese↔English pair was much lower than the two baseline models, reflecting the crucial mismatch between training and testing domains. In contrast, Russian↔English pair benefited the most and achieved surprisingly high BLEU scores. The reason might be due to the proximity of out-of-domain training data and in-domain test data.

[22] http://lotus.kuee.kyoto-u.ac.jp/ASPEC/

[23] http://www.statmt.org/wmt18/translation-task.html

ID	Initialized	Out-of-domain data		In-domain data			BLEU score					
		Ja↔En	Ru↔En	Ja↔Ru	Ja↔En	Ru↔En	Ja→Ru	Ru→Ja	Ja→En	En→Ja	Ru→En	En→Ru
(b3)	-	-	-	✓	✓	✓	3.72	8.35	10.24	12.43	22.10	16.92
I	-	✓	✓	-	-	-	0.00	0.15	4.59	4.15	•25.22	•20.37
II	I	-	-	-	✓	✓	0.20	0.70	•14.10	**•17.80**	•28.23	•24.35
III	I	✓	✓	-	✓	✓	0.23	1.07	•13.31	•17.74	**•28.73**	**•25.22**
IV	II	-	-	✓	-	-	•5.44	•10.67	0.12	3.97	0.11	3.66
V	II	-	-	✓	✓	✓	•6.90	•11.99	•14.34	•16.93	•27.50	•23.17
VI	III	-	-	✓	-	-	•5.91	•10.83	0.26	2.18	0.18	1.10
VII	III	-	-	✓	✓	✓	•7.49	•12.10	**•14.63**	•17.51	•28.51	•24.60
I’	-	✓	✓	✓	✓	✓	•5.31	•10.73	•14.41	•16.34	•27.46	•23.21
II’	I	-	-	✓	✓	✓	•6.30	•11.64	•14.29	•16.83	•27.53	•23.00
III’	I	✓	✓	✓	✓	✓	**•7.53**	**•12.33**	•14.19	•16.77	•27.94	•23.97

Table 3.11: BLEU scores obtained through multistage fine-tuning. “Initialized” column indicates the model used for initializing parameters that are fine-tuned on the data indicated by ✓. **Bold** indicates the best BLEU score for each translation direction. “•” indicates statistical significance of the improvement over (b3).

The first fine-tuning stage significantly pushed up the translation quality for Japanese↔English and Russian↔English pairs, in both cases with fine-tuning (II) and mixed fine-tuning (III). At this stage, both models performed only poorly for Japanese↔Russian pair as they have not yet seen Japanese↔Russian parallel data. Either model had a consistent advantage over the other.

When these models were further fine-tuned only on the in-domain Japanese↔Russian parallel data (IV and VI), we obtained translations of better quality than the two baselines for Japanese↔Russian pair. However, as a result of complete ignorance of Japanese↔English and Russian↔English pairs, the models only produced translations of poor quality for these language pairs. In contrast, mixed fine-tuning for the second fine-tuning stage (V and VII) resulted in consistently better models than conventional fine-tuning (IV and VI), irrespective of the choice at the first stage, thanks to the gradual shift of parameters realized by in-domain Japanese↔English and Russian↔English parallel data. Unfortunately, the translation quality for Japanese↔English and Russian↔English pairs sometimes degraded from II and III. Nevertheless, the BLEU scores still retain a large margin against two baselines.

The last three rows in Table 3.11 present BLEU scores obtained by the methods with fewer fine-tuning steps. The most naive model I’, trained on the balanced mixture of whole five types of corpora from scratch, and the model II’, obtained through a single-step conventional fine-tuning of I on all the in-domain data, achieved only BLEU scores consistently worse than VII. In contrast, when we merged our two fine-tuning steps into a single mixed fine-tuning on I, we obtained a model III' which is better for the Japanese↔Russian pair than VII. Nevertheless, they are still comparable to those of VII, and the BLEU scores for the other two language pairs are much lower than VII. As such, we conclude that our multistage fine-tuning leads to a more robust in-domain multilingual model.

3.5.4 Further Augmentation with Back-translation

Having obtained a better model, we examined again the utility of back-translation. More precisely, we investigated (a) whether the pseudo-parallel data generated by an improved NMT model leads to a further improvement and (b) whether one more stage of fine-tuning on the mixture of original parallel and pseudo-parallel data will result in a model better than training a new model from scratch as examined in Section 3.4.4.

Given an NMT model, we first generated six-way pseudo-parallel data by translating monolingual data. For the sake of comparability, we used the identical monolingual sentences sampled in Section 3.4.4. Then, we further fine-tuned the given model on the mixture of the generated pseudo-parallel data and the original parallel data, following the same over-sampling procedure in Section 3.4.4. We repeated these steps five times.

Table 3.12 shows the results. "new #10" in the second row indicates an M2M Transformer model trained from scratch on the mixture of six-way pseudo-parallel data generated by VII and the original parallel data. It achieved higher BLEU scores than #10 in Table 3.8 thanks to the pseudo-parallel data of better quality but underperformed the base NMT model VII. In contrast, our fine-tuned model VIII successfully surpassed VII, and one more iteration (IX) further improved BLEU scores for all translation directions, except Russian→English. Although further iterations did not necessarily gain BLEU scores, we came to a much higher plateau compared to the results in Section 3.4.

No	Initialized	BT	BLEU score					
			Ja→Ru	Ru→Ja	Ja→En	En→Ja	Ru→En	En→Ru
#10	-	(b3)	4.43	9.38	12.06	14.43	23.09	17.30
new #10	-	VII	•6.55	•11.36	•13.77	•15.59	•24.91	•20.55
VIII	VII	VII	•7.83	•12.21	•15.06	•17.19	•28.49	•23.96
IX	VIII	VIII	•8.03	•12.55	•15.07	•17.80	•28.16	•24.27
X	IX	IX	•7.76	•12.59	•15.08	•18.12	•28.18	•24.67
XI	X	X	•7.85	•12.97	•15.26	•17.83	•28.49	•24.36
XII	XI	XI	•8.16	•13.09	•14.96	•17.74	•28.45	•24.35

Table 3.12: BLEU scores achieved through fine-tuning on the mixture of the original parallel data and six-way pseudo-parallel data. The "Initialized" column indicates the model used for initializing parameters, and so does the "BT" column the model used to generate pseudo-parallel data. "•" indicates the statistical significance of the improvement over #10.

3.5.5 Summary

We conducted a throughout comparison of the existing methods on our target task using only in-domain data. However, experiment results showed the limitations of using just restricted in-domain data. Therefore we proposed a multistage fine-tuning approach to practically involve large-scale out-of-domain data.

Table 3.13 summarizes the progression of BLEU scores at each investigation step over our in-domain data.

Investigation step	Ja→Ru	Ru→Ja
Uni-directional Transformer: (b1) in Table 3.5	0.70	1.96
M2M Transformer: (b3) in Table 3.5	3.72	8.35
+ six-way pseudo-parallel data: #10 in Table 3.8	4.43	9.38
M2M multistage fine-tuning: VII in Table 3.11	7.49	12.10
+ six-way pseudo-parallel data: XII in Table 3.12	8.16	13.09

Table 3.13: Summary of our investigation: BLEU scores of the best NMT systems at each step.

We demonstrated that incorporating multistage multilingual domain adaptation significantly boost the performance on our language pair of interest (VII and XX).

3.6 Leveraging In-domain Data From Other Languages

In this chapter, we focus on only the news domain of additional Japanese↔English and Russian↔English auxiliary parallel corpora, which we will refer to as pivot parallel corpora. We investigate how translation results are improved by using in-domain pivot parallel corpora (Japanese↔English and Russian↔English) in M2M Transformer (also see Section 3.4) modeling.

3.6.1 Experimental Settings

Data

To train M2M Transformer systems, we used the news domain data provided by WAT2019[24]. More specifically, we used Global Voices[25] as training data for Japanese↔Russian, Japanese↔English and Russian↔English, and manually aligned, cleaned, and filtered News Commentary data was used as development and test sets.[26] Additionally, we utilized Jiji[27] and News Commentary[28] data for Japanese↔English and Russian↔English, respectively. Table 3.14 summarizes the size of train/development/test splits used in our experiments.

Lang.pair	Source	Partition	#sent.	#tokens	#types
Ja↔Ru	Global Voices	train	12,356	341k / 229k	22k / 42k
	News Commentary	development	486	16k / 11k	2.9k / 4.3k
	News Commentary	test	600	22k / 15k	3.5k / 5.6k
Ja↔En	Global Voices	train	47,082	1.27M / 1.01M	48k / 55k
	Jiji	train	200,000	5.84M / 5.11M	45k / 78k
	News Commentary	development	589	21k / 16k	3.5k / 3.8k
Ru↔En	Global Voices	train	82,072	1.61M / 1.83M	144k / 74k
	News Commentary	train	279,307	7.00M / 7.41M	214k / 89k
	News Commentary	development	313	7.8k / 8.4k	3.2k / 2.3k

Table 3.14: Statistics on our in-domain parallel data.

[24] http://lotus.kuee.kyoto-u.ac.jp/WAT/WAT2019/index.html

[25] https://globalvoices.org/

[26] https://github.com/aizhanti/JaRuNC

[27] http://lotus.kuee.kyoto-u.ac.jp/WAT/jiji-corpus/

[28] http://lotus.kuee.kyoto-u.ac.jp/WAT/News-Commentary/news-commentary-v14.en-ru.filtered.tar.gz

We tokenized English and Russian sentences using *tokenizer.perl* of `Moses` (Koehn et al., 2007).[29] To tokenize Japanese sentences, we used `MeCab`[30] with the IPA dictionary. After tokenization, we eliminated duplicated sentence pairs and sentences with more than 100 words for all the languages.

Systems

This section describes our MultiCorpora system and our baseline, which are based on the same M2M Transformer architecture (Johnson et al., 2017) but trained on different training corpora (Table 3.14). Here, M2M Transformer translates from multiple source languages into different target languages within a single model. Since we have 3 language pairs, we concatenate all pairs in both directions with over-sampling to match the biggest parallel data. We add a target language token to the source side of each pair and treat it like a single language-pair case.

We experiment with the following systems:

- **MultiCorpora**: Our system is trained on a balanced concatenation of Global Voices, Jiji, and News Commentary corpora on 6 translation directions.
- **Only GV**: This is our baseline system, which is trained on only Global Voices data on 6 translation directions, the same as in Imankulova et al. (2019).

Only GV is used as a comparative model to investigate the effect of additional pivot corpora.

Although we train our models on 6 translation directions, we only report the BLEU scores on Japanese→Russian and Russian→Japanese test sets.

3.6.2 Results

Table 3.15 demonstrates the BLEU scores of our baseline Only GV model and proposed MultiCorpora model on News Commentary Japanese→Russian[31] and Russian→Japanese[32] test data for News Commentary shared task. Our MultiCorpora

[29] https://github.com/moses-smt/mosesdecoder

[30] http://taku910.github.io/mecab, version 0.996.

[31] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/list.php?t=66o=4

[32] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/list.php?t=67o=1

Models	Ja→Ru	Ru→Ja
Only GV	3.66	8.79
MultiCorpora	**6.59**	**11.00**

Table 3.15: Evaluation results: BLEU scores. **Bold** indicates the best BLEU score for each translation direction.

system trained on additional pivot parallel corpora exceeded the baseline Only GV model trained without additional pivot parallel corpora by approximately 3 BLEU points on both Japanese→Russian and Russian→Japanese.

3.6.3 Discussion

We investigate the effect of adding Jiji and News Commentary corpora as pivot parallel corpora to original Global Voices training data. In extremely low-resource machine translation in the news domain, unknown tokens become a serious issue due to vocabulary coverage. Adding the pivot parallel corpora to training data can be expected to increase vocabulary coverage.

Therefore, we investigate how much vocabulary coverage was improved by using pivot parallel corpora. For that purpose, we investigate the following vocabulary sets $\mathcal{A}$ and $\mathcal{B}$:

$$\mathcal{A} = \mathcal{T} \cap \mathcal{G} \tag{3.1}$$

$$\mathcal{B} = \mathcal{T} \cap (\mathcal{G} \cup \mathcal{P}) \tag{3.2}$$

$\mathcal{T}$ is a set of unknown tokens from test data not included in the direct Japanese↔Russian 12k training data, $\mathcal{G}$ is pivot Global Voices vocabulary set, and $\mathcal{P}$ is Jiji and News Commentary training vocabulary set. By comparing the number of tokens and types of distinct words of $\mathcal{A}$ and $\mathcal{B}$, you can see how much the coverage has increased. In addition, we investigate how correctly the tokens added by the Jiji corpus, and News Commentary are translated. If a token from the vocabulary set of $\mathcal{A}$ or $\mathcal{B}$ appeared in both the gold sentence and the translated sentence of the system, it was counted as being correctly translated.

Table 3.16 shows token and type coverage and correctly translated tokens and types of distinct words on test data for $\mathcal{A}$ and $\mathcal{B}$, respectively. It can be seen that

	Ja→Ru				Ru→Ja			
	$\mathcal{A}$ (Only GV)		$\mathcal{B}$ (MultiCorpora)		$\mathcal{A}$ (Only GV)		$\mathcal{B}$ (MultiCorpora)	
	#tokens	#types	#tokens	#types	#tokens	#types	#tokens	#types
Coverage in data	1,467	1,220	2,072	1,751	481	362	596	450
Correctly translated	85	65	191	147	26	21	31	24

Table 3.16: The coverage of tokens from additional pivot parallel data and the number of correctly translated tokens and types of distinct words by each system calculated for the test set.

both Russian and Japanese have improved $\mathcal{B}$ coverage compared to $\mathcal{A}$. In particular, the coverage of Russian is greatly improved. And by adding Jiji corpus and News Commentary to the training data, you can see that the number of correctly translated tokens has increased. This shows that vocabulary coverage has increased and translation accuracy has improved. On the other hand, the number of correctly translated tokens is few compared to increased coverage from additional parallel data. This is considered to be due to difficulty of directly learning Japanese↔Russian translation from added indirect Japanese↔English and Russian↔English pivot corpora.

Furthermore, in order to deepen the knowledge about the tokens covered using pivot corpora, we analyze the cases where the newly added tokens by Jiji and News Commentary corpora are translated correctly and incorrectly. By adding Jiji and News Commentary corpora, we define the vocabulary set newly covered by the test data vocabulary as $\mathcal{C}$ as follows:

$$\mathcal{C} = (\mathcal{T} \cap \mathcal{P}) - \mathcal{G} \tag{3.3}$$

Table 3.17 shows translation examples of only GV and MultiCorpora systems. The [unknown tokens] in each sentence belong to $\mathcal{C}$. The first sentence is an example (a) where MultiCorpora was able to correctly translate “株主” compared to Only GV. On the other hand, the second example shows that neither MultiCorpora nor Only GV could correctly translate an unknown token “表立つ” included in pivot parallel corpora. It is considered that it cannot be translated because the whole sentence was translated incorrectly.

(a)	**Source**	Должны ли акционеры быть королями ?
	Target	[株主] が、王様になるべきか? (Should [shareholders] be kings ?)
	Only GV	この акционер が社会の中心となっているのだろうか？ (Is this акционер the center of society?)
	MultiCorpora	[株主] は王を持つべきなのか？ (Should [shareholders] have a king?)
(b)	**Source**	Преемственность всегда оставалась сугубо семейным делом , и все споры оставались за закрытыми дверями .
	Target	これまで、継承者は、厳格に首長家から選ばれるものとされ、いかなる論争も [表立っ] てされることはなかった。 (The succession was always strictly a family affair , and no disputes have [emerged].)
	Only GV	家族経営のドライクリーニング店で、常習的な商事には至っていない。 (It is a family-run dry cleaning shop, and it has not become a regular business.)
	MultiCorpora	このような虐待は日々くり返されていた。 (Such abuse was repeated every day.)

Table 3.17: Examples of translating [unknown tokens] included in pivot parallel data $\mathcal{C}$ from Russian into Japanese.

3.6.4 Summary

The difficult part of Japanese↔Russian news translation task is unknown tokens due to difficult news domain covering various topics and extremely low-resource available parallel data. To address this issue, we investigated the coverage of translatable tokens by training M2M Transformer using an in-domain pivot parallel corpora. As a result, we found out that our system MultiCorpora, can translate more tokens by taking advantage of additional pivot parallel corpora.

4 | Additional Modality for Low-Resource NMT

4.1 Introduction

Generally, studies tackle the low-resource simultaneous translation problem of incomplete input by optimizing the timing of reading and translating the input text. On the other hand, in this study, we aim to improve the quality of translation by using additional modalities, in this case, visual modality. The situation where an image is given as an input is specific as a setting for simultaneous translation. However, if we demonstrate that image information is effective for simultaneous translation, humans can perform simultaneous translation using visual information as well. We hope that this will lead to the development of such systems and the development of simultaneous translation using information outside the given text.

To this end, we propose Multimodal Simultaneous Neural Machine Translation (**MSNMT**) that supplements the incomplete textual modality with visual information, in the form of an image. It will predict still missing information to improve translation quality during the decoding process.

Our approach in the future can be applied in various situations where visual information is related to the content of speech, such as presentations with slides (e.g., TED Talks[1]) and news video broadcasts[2]. Our experiments show that the proposed MSNMT method achieves higher translation accuracy than the SNMT model that does not use images by leveraging image information. To the best of our knowledge, we are the first to propose the incorporation of visual information to solve the problem of incomplete text information in SNMT.

The main contributions are as follows:

- We propose to combine multimodal and simultaneous NMT, therefore, discovering cases where such multimodal signals are beneficial for the end-task. Our MSNMT approach brings significant improvement in the quality of low-resource simultaneous translation by enriching incomplete text input information using visual clues.

[1] https://interactio.io/

[2] https://www.a.nhk-g.co.jp/bilingual-english/broadcast/nhk/index.html

- As a result of a thorough analysis, we conclude that the proposed method is able to predict tokens that have not appeared yet for source-target language pairs with different word order (e.g., English→Japanese).
- By providing an adversarial evaluation, we showed that the models indeed utilize visual information.

4.2 Related Work

For simultaneous translation, it is crucial to predict the words that have not appeared yet. For example, it is important to distinguish nouns in SVO-SOV translation and verbs in SOV-SVO translation (Ma et al., 2019). SNMT can be implemented with two types of policy: fixed and adaptive policies (Zheng et al., 2019a). Adaptive policy decides whether to wait for another source word or emit a target word in one model. Previous models with adaptive policies include explicit prediction of the sentence-final verb (Grissom II et al., 2014; Matsubara et al., 2000) and unseen syntactic constituents (Oda et al., 2015). Most dynamic models with adaptive policies (Gu et al., 2017; Dalvi et al., 2018; Arivazhagan et al., 2019; Zheng et al., 2019b,c, 2020) have the advantage of exploiting input text information as effectively as possible due to the lack of such information in the first place. Meanwhile, Ma et al. (2019) proposed a simple `wait-k` method with fixed policy, which generates the target sentence only from the source sentence that is delayed by `k` tokens. However, their model for simultaneous translation relies only on the source sentence. In this research, we concentrate on the `wait-k` approach with the fixed policy so that the amount of input textual context can be controlled to better analyze whether multimodality is effective in SNMT.

Multimodal NMT (MNMT) for full-sentence machine translation has been developed to enrich text modality by using visual information (Hitschler et al., 2016; Specia et al., 2016; Elliott and Kádár, 2017). While the improvement brought by visual features is moderate, their usefulness is proven by Caglayan et al. (2019). They showed that MNMT models are able to capture visual clues under limited textual context, where source sentences are synthetically degraded by color deprivation, entity masking, and progressive masking. However, they use an artificial setting where they deliberately deprive the models of source-side textual context by masking. However, our research has discovered an actual end-task and has shown the effectiveness of using multimodal data for it. Compared with the entity masking experiments (Caglayan et al., 2019), where they use a model exposed to only k words, our model starts by waiting for the first k source words and then generates each target word after receiving every new source token, eventually seeing all input text.

In MNMT, visual features are incorporated into standard machine translation in many ways. Doubly-attentive models are used to capture the textual and visual context vectors independently and then combine these context vectors in a concatenation manner (Calixto et al., 2017) or hierarchical manner (Libovický and Helcl, 2017). Some studies use visual features in a multitask learning scenario (Elliott and Kádár, 2017; Zhou et al., 2018). Also, recent work on MNMT has partly addressed lexical ambiguity by using visual information (Elliott et al., 2017; Lala and Specia, 2018; Gella et al., 2019) showing that using textual context with visual features outperform unimodal models.

In our study, visual features are extracted using image processing techniques and then integrated into an SNMT model as additional information, which is supposed to be useful to predict missing words in a simultaneous translation scenario. To the best of our knowledge, this is the first work that incorporates external knowledge into an SNMT model.

4.3 Multimodal Simultaneous Neural Machine Translation Architecture

Our main goal is to investigate if image information would bring improvement to a low-resource SNMT. As a result, two tasks could benefit from each other by combining them.

In this section, we describe our MSNMT model, which is composed by combining an SNMT framework `wait-k` (Ma et al., 2019) and a multimodal model (Libovický and Helcl, 2017) (Figure 4.1, MSNMT (`wait-k`)). We base our model on the RNN architecture, which is widely used in MNMT research (Libovický and Helcl, 2017; Caglayan et al., 2017a; Elliott and Kádár, 2017; Zhou et al., 2018;

Hirasawa et al., 2019). The model takes a sentence and its corresponding image as inputs. The decoder of the MSNMT model outputs the target language sentence in a simultaneous and multimodal manner by attaching attention not only to the source sentence but also to the image related to the source sentence.[3]

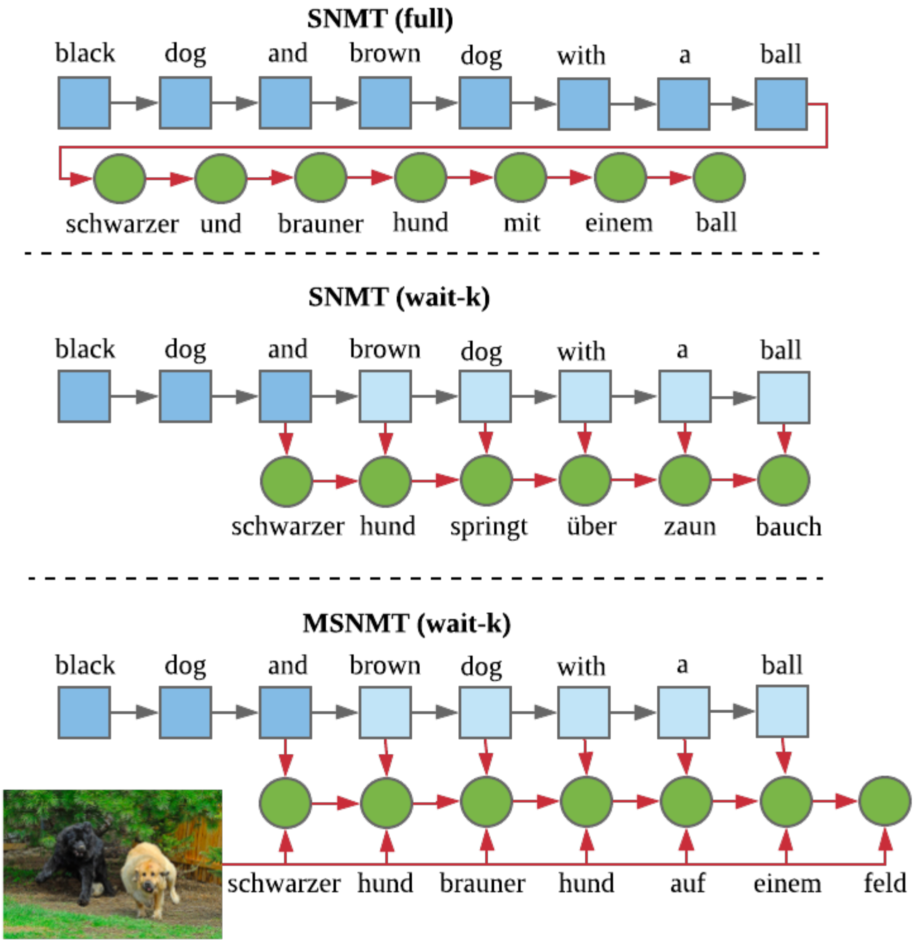


Figure 4.1: Example of multimodal simultaneous machine translation based on `wait-k` approach incorporating visual clues for better English→German translation.

[3] Our code is publicly available at: https://github.com/toshohirasawa/mst.

4.3.1 Simultaneous Translation

We first briefly review standard NMT to set up the notations (see also Figure 4.1, SNMT (full)). The encoder of standard NMT model always takes the whole input sequence $\mathbf{X} = (x_1, ..., x_n)$ of length n where each x_i is a word embedding and produces source hidden states $\mathbf{H} = (h_1, ..., h_n)$. The decoder predicts the next output token y_t using $\mathbf{H}$ and previously generated tokens, denoted $\mathbf{Y}_{<t} = (y_1, ..., y_{t-1})$. The final output is calculated using the following equation:

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{|\mathbf{Y}|} p(y_t|\mathbf{X}, y_{<t}) \tag{4.1}$$

Different from standard neural translation, in which each y_i is predicted using the entire source sentence $\mathbf{X}$, the simultaneous translation requires the model to translate concurrently with the growing source sentence. We incorporate the `wait-k` approach (Ma et al., 2019) for our simultaneous translation model (Figure 4.1, SNMT (`wait-k`)). Instead of waiting for the whole sentence before translating, this model waits for only the first `k` tokens and starts to generate each target tokens after taking every new source token one by one. It stops taking new input tokens once the whole input sentence is on board. For example, if $\texttt{k} = 3$, the first target token is predicted using the first 3 source tokens, and the second target token using the first 4 source tokens. The `wait-k` decoding probability $p_{\texttt{wait-k}}$ is:

$$p_{\texttt{wait-k}}(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{|\mathbf{Y}|} p(y_t|\mathbf{X}_{\leq g(t)}, y_{<t}) \tag{4.2}$$

Where $g(t)$ is the `wait-k` policy function which decides how much input text to read and translate, $\mathbf{X}_{\leq g(t)} = (x_1, ..., x_{g(t)})$. $g(t)$ is defined as follows:

$$g(t) = \text{min}\{k + t - 1, n\} \tag{4.3}$$

When $k+t-1$ is over source length n, $g(t)$ is fixed to n, which means the remaining target tokens (including current step) are generated using the full source sentence. For full sentence translation, $g(t)$ is constant $g(t) = n$.

4.3.2 Multimodal Translation

We use a hierarchical attention combination technique (Libovický and Helcl, 2017) to incorporate visual and textual features into an MNMT model. This model calculates the independent context vectors from the textual features $\boldsymbol{h}^{\text{txt}} = (h_1^{\text{txt}}, ..., h_n^{\text{txt}})$ and the visual features $\boldsymbol{h}^{\text{img}} = (h_1^{\text{img}}, ..., h_m^{\text{img}})$, which are extracted by the textual encoder and the image processing model, respectively. It then combines the resulting two vectors using a second attention mechanism, which helps to perform simultaneous translation taking into account visual information.

Specifically, we compute the context vectors c_i^{f} for each image (f $=$ img) and text (f $=$ txt) modality independently using the following equations:

$$
\begin{aligned}
e_{i,j}^{\text{f}} &= \Omega^{\text{f}}(s_i, h_j^{\text{f}}) && (4.4)\\
\alpha_{i,j}^{\text{f}} &= \frac{\exp(e_{i,j}^{\text{f}})}{\sum_{l=1}^{|\mathbf{h}^{\text{f}}|} \exp(e_{i,l}^{\text{f}})} && (4.5)\\
c_i^{\text{f}} &= \sum_{j=1}^{|\mathbf{h}^{\text{f}}|} \alpha_{i,j}^{\text{f}} h_j^{\text{f}} && (4.6)
\end{aligned}
$$

where Ω^{f} is a feedforward network for each modality f; s_i is i-th decoder hidden state.

We project these image and text context vectors into a common space and compute another distribution over the projected context vectors and their corresponding weighted average using the second attention:

$$
\begin{aligned}
\tilde{e}_i^{\text{f}} &= \Psi(s_i, c_i^{\text{f}}) && (4.7)\\
\beta_i^{\text{f}} &= \frac{\exp(\tilde{e}_i^{\text{f}})}{\sum_{\text{r}\in\{\text{img},\text{txt}\}} \exp(\tilde{e}_i^{\text{r}})} && (4.8)\\
\tilde{c}_i &= \sum_{\text{r}\in\{\text{img},\text{txt}\}} \beta_i^{\text{r}} W^{\text{r}} c_i^{\text{r}} && (4.9)
\end{aligned}
$$

where Ψ is a feedforward network. Equation 4.8 calculates the second attention to combine the image and text vectors. W^{r} is a weight matrix used to compute the context vector $\tilde{c}_i$ calculated from image and text features. The final hypothesis $\mathbf{Y}$

has the probability:

$$p_{\text{mnmt}}(\mathbf{Y}|\mathbf{X},\mathbf{Z}) = \prod_{t=1}^{|\mathbf{Y}|} p(y_t|\mathbf{X},\mathbf{Z},y_{<t}) \tag{4.10}$$

where **Z** represents input image features.

4.3.3 Multimodal Simultaneous Neural Machine Translation

In this subsection, we describe the structure of the MSNMT model, which is a combination of the models described in Sections 4.3.1 and 4.3.2. The method for calculating the image context vector is the same as for MNMT; however, the text context vector (Equation 4.6) for the t-th step is calculated as follows:

$$\hat{c}_i^{\text{txt}} = \sum_{j=1}^{g(t)} \alpha_{i,j}^{\text{txt}} h_j^{\text{txt}} \tag{4.11}$$

Thus $\hat{c}_i^{\text{txt}}$ is calculated from the input text prefix determined by `wait-k` policy function $g(t)$. Then we apply the second attention to $\hat{c}_i^{\text{txt}}$ and c_i^{img} in order to calculate $\tilde{c}_i$ (Equation 4.9).

The decoding probability becomes as follows:

$$p_{\text{msnmt}}(\mathbf{Y}|\mathbf{X},\mathbf{Z}) = \prod_{t=1}^{|\mathbf{Y}|} p(y_t|\mathbf{X}_{\leq g(t)},\mathbf{Z},y_{<t}) \tag{4.12}$$

4.4 Experimental Setup

4.4.1 Dataset

We experiment with our model in four translation directions consisting of 5 languages: English (En), German (De), French (Fr), Czech (Cs), and Japanese (Ja). All language pairs include English on the source side.

We used the train, development, and test sets from the Multi30k (Elliott et al., 2016) dataset published in the WMT16 Shared Task, which is a benchmark dataset generally used in MNMT research (Libovický and Helcl, 2017; Caglayan et al., 2019; Elliott and Kádár, 2017; Zhou et al., 2018; Hirasawa et al., 2019) for English→German, English→French and English→Czech.

Nakayama et al. (2020) released F30kEnt-JP dataset[4] which contains Japanese translations of first two original English captions for each image of the Flickr30k Entities dataset (Plummer et al., 2017). They follow the same annotation rules as the Flickr30k Entities dataset using exactly the same tags with entity types and IDs.

We preprocessed this data as follows: 1) The parallel English→Japanese data was created by taking alignment using corresponding IDs assigned to each Japanese translation entities with the IDs of Flickr30k entities.[5] 2) The created parallel data was aligned with its corresponding images using text files named $(image\_id).txt$ corresponding to each image in Flickr30k. 3) Finally, the created multimodal data was split to train, dev, and test following data splits of Multi30k using the same Multi30k image IDs. Note that the English side of English→Japanese parallel data extracted from F30kEnt-JP and English side of Multi30k data are thought to be somewhat comparable but not strictly the same while their corresponding images are the same.

Data splits and average sentence length for each language are shown in Table 4.1.

Parallel data	#sent.	Avg. sent. length				
		English	German	French	Czech	Japanese
Train	29,000	13	12	14	10	20
Dev	1,014	13	12	14	10	20
Test	1,000	12	12	13	10	20

Table 4.1: Dataset statistics.

We limit the vocabulary size of the source and the target languages after concatenating them to 10,000 sub-words (Sennrich et al., 2016b). All sentences are preprocessed with lower-casing, tokenizing, and normalizing the punctuation using the Moses script[6]. To tokenize Japanese sentences, we used MeCab[7] with the IPA

[4] https://github.com/nlab-mpg/Flickr30kEnt-JP

[5] We used the second translations due to some empty translations of the first captions.

[6] We applied preprocessing using task1-tokenize.sh from https://github.com/multi30k/dataset.

[7] http://taku910.github.io/mecab, version 0.996.

dictionary.

Visual features are extracted using pre-trained ResNet (He et al., 2016b). Technically, we encode all images in Multi30k with ResNet-50 and pick out the hidden state in the pool5 layer as a 2,048-dimension visual feature.

4.4.2 Systems

We compare the following models: **1. SNMT:** We use only text modality for training data as a baseline for each `wait-k` model. **2. MSNMT:** We use image modality along with text modality for training data for each `wait-k` model.

To train the above models, we utilize attention NMT (Bahdanau et al., 2014) with a 2-layer unidirectional GRU encoder and a 2-layer conditional GRU decoder. We use the open-source implementation of the `nmtpytorch` toolkit v3.0.0 (Caglayan et al., 2017b). We first pre-train the MSNMT model for each `k` until convergence using only text data and use zeros for visual features. Then we continue training MSNMT on multimodal data for each `k`. We employ early-stopping: the training was stopped when the BLEU score did not increase on the development set for 10 epochs for MSNMT pre-training, 5 epochs for MSNMT fine-tuning, and 15 epochs for SNMT training.

In order to keep our experiments as pure as possible, we will not use additional data or other types of models. It will allow us to control the amount of input textual context, so we can easily analyze the relationship between the amount of textual and visual information.

4.4.3 Hyperparameters

We use the same hyperparameters for SNMT and MSNMT for a fair comparison as follows. All models have word embeddings of 200 and recurrent layers of dimensionality 400 units with 2way sharing of embeddings in the network. We used Adam (Kingma and Ba, 2015) with a learning rate of 0.0004. Decoders were initialized with zeros. We used a minibatch size of 64 for training and 32 for fine-tuning. Rates of dropout applied on source embeddings, source encoder states, and pre-softmax activations were 0.4, 0.5, and 0.5, respectively. We set the max length of the input to 100. wait-k experiments were conducted for 1, 3, 5, 7, and Full settings. For MSNMT only hyperparameters, sampler type was set to approximate, and channels were set to 2048. The fusion type was set to hierarchical mode.

4.4.4 Evaluation

We report BLEU scores, which is a widely used evaluation metric in MT, on our test sets for each wait-k model.[8] Note that reported BLEU scores were calculated using the whole generated target sentence. Statistical significance ($p < 0.05$) on the difference of BLEU scores was tested by Moses's *bootstrap-hypothesis-difference-significance.pl*. "Full" means that the whole input sentence is used as an input for the model to start translating. All reported results are the average of four runs using four different random seeds.

Additionally, we use the open-sourced Average Lagging (AL) latency metric proposed by Ma et al. (2019) to evaluate the latency for SNMT and MSNMT systems.

4.5 Results

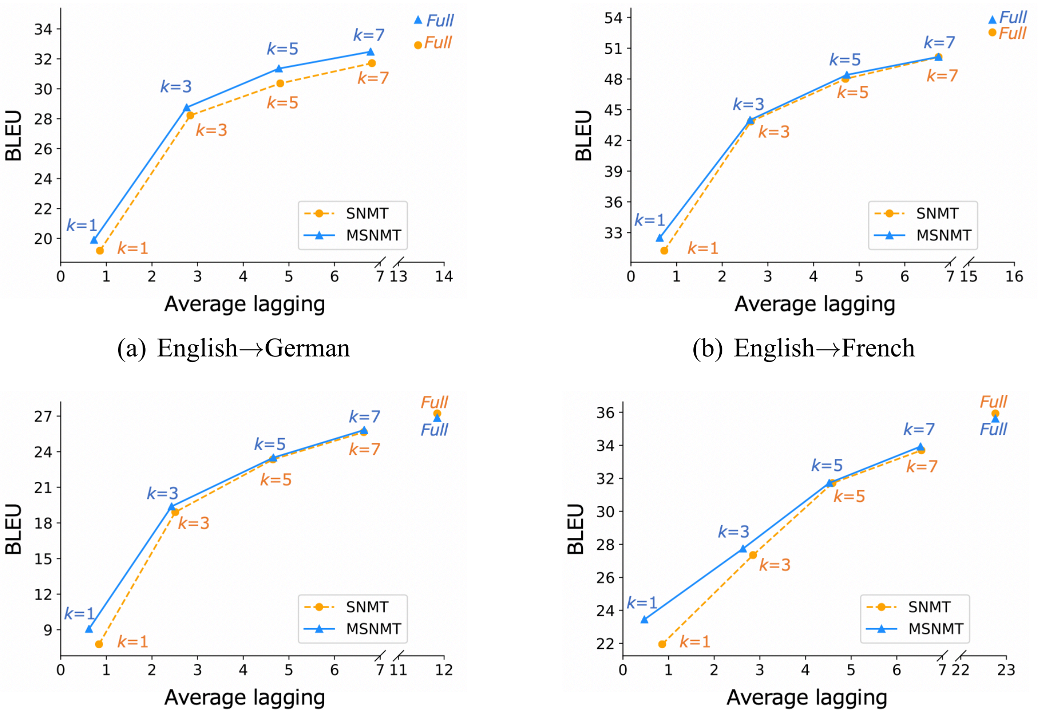
Table 4.2 illustrates the BLEU scores of MSNMT and SNMT models on the test set. For all language pairs, MSNMT systems show significant improvements over SNMT systems when input textual information is limited. Note that the difference of BLEU scores between MSNMT and SNMT becomes larger as the k gets smaller, especially when the target language is distant from English in terms of word order (e.g., Czech and Japanese). On the other hand, the availability of more tokens during the decoding process ($\mathtt{k} \geq 5$) leads to the text information becoming sufficient in some cases.

Figure 4.2 shows translation quality against AL for four language directions. In all these figures, we observe that, as k increases, the gap between BLEU scores for MSNMT and SNMT decreases. We also observe that AL scores are better for MSNMT as k decreases. From these results, it can be seen that in terms of latency, the smaller k is, the more beneficial the visual clues become.

[8] Due to space constraints, we show results only for test sets.

wait-k	En→De		En→Fr		En→Cs		En→Ja	
	S	M	S	M	S	M	S	M
1	19.18	**•19.90**	31.23	**•32.49**	7.78	**•9.07**	21.95	**•23.45**
3	28.22	**•28.75**	43.85	**43.99**	18.91	**•19.39**	27.35	**•27.74**
5	30.38	**•31.48**	48.01	**•48.40**	23.35	**23.50**	31.71	**31.72**
7	31.72	**32.14**	50.14	**50.16**	25.65	**25.83**	33.70	**33.93**
Full	34.64	**34.84**	53.55	**53.78**	**27.22**	26.85	**35.93**	35.62

Table 4.2: BLEU scores of SNMT (S) and MSNMT (M) models for four translation directions on test set. Results are the average of four runs. **Bold** indicates the best BLEU score for each `wait-k` for each translation direction. "•" indicates statistical significance of the improvement over SNMT.



(a) English→German

(b) English→French

(c) English→Czech

(d) English→Japanese

Figure 4.2: Average Lagging scores. Results are the average of four runs.

4.6 Analysis

In this section, we provide a thorough analysis to further investigate the effect of visual data to produce a simultaneous translation by (a) providing adversarial evaluation; and (b) analyzing the effect of different word order for English→Japanese language pair.

4.6.1 Adversarial Evaluation

In order to determine whether MSNMT systems are aware of the visual context (Elliott, 2018), we perform the adversarial evaluation on the test set. We present our system with correct visual data with its source sentence (Congruent) as opposed to random visual data as an input (Incongruent) (Elliott, 2018). For that purpose, we reversed the order of 1,000 images of the test set, so there will be no overlapping congruent visual data. Then we reconstruct image features for those images to use as an input to a model.

Results of image awareness experiments are shown in Table 4.3. We can see the large difference in BLEU scores between MSNMT congruent (C columns) and incongruent (I columns) settings when `k` are small. This implies that our proposed model utilizes images for translation by learning to extract needed information from visual clues. The interesting part is for a full translation, where scores for the incongruent setting outperform or are very close to those of the congruent setting. The reason is that when textual information is enough, visual information becomes not that relevant in some cases.

`wait-k`	En→De		En→Fr		En→Cs		En→Ja	
	C	I	C	I	C	I	C	I
1	**19.90**	8.19	**32.49**	18.00	**9.07**	6.83	**23.45**	17.57
3	**28.75**	26.78	**43.99**	42.31	**19.39**	18.78	**27.74**	24.51
5	**31.48**	31.08	**48.40**	48.19	**23.50**	22.81	**31.72**	28.57
7	**32.14**	32.04	**50.16**	50.15	**25.83**	25.09	**33.93**	31.03
Full	**34.84**	34.40	**53.78**	53.10	**26.85**	26.84	**35.62**	35.59

Table 4.3: Image Awareness results on test set. BLEU scores of MSNMT Congruent (C) and Incongruent (I) settings for four translation directions. Results are the average of four runs. **Bold** indicates the best BLEU score for each `wait-k` for each translation direction.

4.6.2 How Source-Target Word Order Affects Translation

In `wait-k` translations, for the English→Japanese language pair with different word orders (SVO vs. SOV), some source tokens should be translated before they are presented to the decoder for grammaticality and fluency purposes. Hence, the model also needs to handle such cases well apart from the "usual" order. We hypothesize that MSNMT models given additional visual information are able to translate such cases better than SNMT models. Therefore, we investigated how many tokens were correctly translated that are not given as input yet.

t	1	2	3	4	5	6	7	8	9	10	11							
Source	a	person	rappelling	a	cliff	above	a	body	of	water	.							
Target, k=3				海	の	上	に	ある	断崖	を	降り	て	いる	一	人	の	男性	。
Entity count				✓					✗						✗			

Table 4.4: Example of English→Japanese translation to count entities that should be translated before introducing it to a model in case of `wait-3` (see Figure 4.3(a)). A `wait-k` model starts translating after `k` tokens are inputted. Colors represent the same entities. ✓ indicates entities that are not presented to the model at timestep t yet, and ✗ indicates entities that are already seen by the model at timestep t. We count only those entities marked with ✓ for # total entities (Table 4.5).

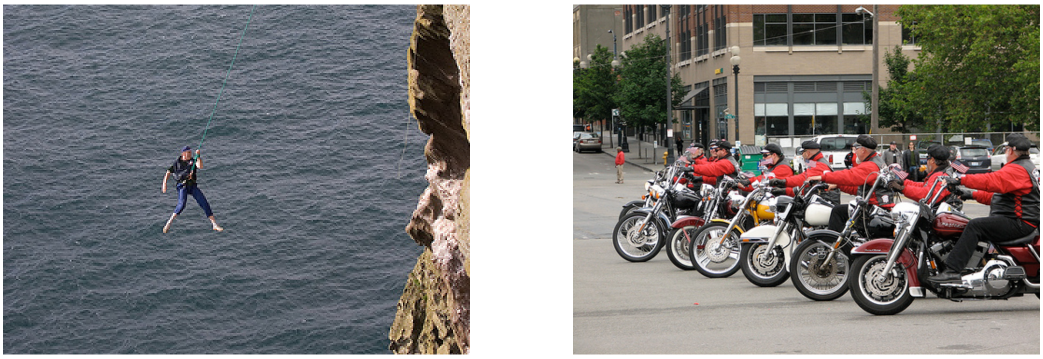
First, we quantitatively analyze how well we can translate entities that are not presented from the source yet but should exist in target sentences. To align the source and target entities, we use the annotation of the entities attached to both the source and target sentences. Given that annotated entities have the same IDs and tags for both English and Japanese, we can align, calculate, and extract those entities from source and target sentences. If the index of the first token of the aligned target entity is not given as input at timestep `k` yet, we count them for each `k` scenario as # total entities (Table 4.5). For example, in Table 4.4 a `wait-3` model should start translating after a token "rapelling" is presented to the model. And if an ID of the entity of "海 (a body of water)" is in the target sentences but not in the inputted part yet, we count it as an entity that should be translated before being inputted into the model. Similarly, an entity of "断崖 (cliff)" is already presented to the model at timestep 5, so we do not count those entities. If the same entity ID appears more than once in one sentence, we exclude those entities due to the impossibility of alignments. Finally, for each model during decoding, if those entities are included in the translation results of the model with a perfect match from pre-calculated # total entities, we consider them as correctly translated.[9]

[9] We can not create # total entities from decoded tokens directly due to unavailability of entity annotations.

k	# total entities	# correct entities by S		# correct entities by M	
		wait-k	Full	wait-k	Full
1	1,343	251	**716**	**270**	707
3	852	229	**433**	**242**	432
5	502	147	**247**	**151**	243
7	320	106	**160**	106	159

Table 4.5: Number of entities that were correctly translated before being presented to the model by SNMT (S) and MSNMT (M) models with their for each k. Results are the average of four runs.

Table 4.5 demonstrates the results. k column is to determine how many tokens a model waits before starting translating. Note that k=Full is not included because all entities are given at the time of translation. The reason that the total number of entities that were not inputted yet decreases when k increases (# total entities column) is that more entities are already available for the model for translation. wait-k columns show how many entities were correctly translated by wait-k SNMT and MSNMT models from # total entities for each k scenario. Columns Full show upper-bounds of how many entities can be correctly translated if the models were trained with full sentences for entities from each k. Comparing Full results to wait-k for both SNMT and MSNMT shows that it is hard to correctly translate entities when k is small. Furthermore, comparing wait-k results of SNMT to MSNMT, it can be seen that the smaller value of k, the better MSNMT can handle different source-target word order than SNMT.



(a) A person rappelling a cliff.

(b) Eight men on motorcycles.

Figure 4.3: Images presented in translation examples (Table 4.6).

Source	a person rappelling a cliff above a body of water .
Target	海の上にある断崖を降りている一人の男性。
S `wait-3`	誰かが、岩の上で崖を登る。(someone climbs a cliff on a rock.)
M `wait-3`	人が海の上で崖を降りている。(a person is rappelling a cliff above the sea.)
S `Full`	人が水域の上の崖を登っている。(a person is climbing a cliff above a body of water.)
M `Full`	人が水域の上で崖を降りている。(a person is rappelling a cliff above a body of water.)
Source	eight men on motorcycles dressed in red and black are all lined up on the side of the street .
Target	赤と黒の服を着たオートバイに乗っている8人の男性が通りの脇にずらりと並んでいる。
S `wait-3`	白い服を着て、黒と黒の服を着た1人の男性が、通りの脇に並んでいる。 (a man in white and black and black is standing beside the street.)
M `wait-3`	自転車に乗っている赤と黒の服を着た8人の男性が、通りの側面にある。 (eight men in red and black clothes riding a bicycle are on the side of the street.)
S `Full`	赤と黒の服を着た、オートバイに乗った2人の男性が、通りの脇で並んでいる。 (two men on motorcycles, dressed in red and black, line up by the side of the street.)
M `Full`	赤と黒の服を着た、オートバイに乗った8人の男性が、通りの側面に並んでいる。 (eight men on motorcycles, dressed in red and black, line the side of the street.)

Table 4.6: Examples of English→Japanese translations from test set using SNMT (S) and MSNMT (M) models (also refer to Figure 4.3). In () are shown their English meanings. The same colors indicate the same entity types.

As an example, we sampled sentences and their images from English→Japanese test set (Figure 4.3) to compare the outputs of our systems. Table 4.6 lists their translations generated by SNMT (S) and MSNMT (M) models. In the first example, an SNMT model with `wait-3` could not predict "海 (sea, a body of water)" which appears at the end of the source sentence and generated an erroneous "岩 (rock)" which is not present neither in source text nor in a corresponding image. Contrarily, the MSNMT model with `wait-3` was able to correctly predict "海 (body of water)" even before it was inputted by capturing visual information. When a full sentence is given as an input, MSNMT translated it correctly using more information, unlike SNMT, which translated only from the given text and generated incorrect "登って (climbing)" instead of "降りて (rappelling)". Interestingly, in the second example, the MSNMT model with `wait-3` predicted "自転車 (bicycles)" instead of "オートバイ (motorcycles)" at the beginning of the sentence, while the SNMT model with `wait-3` was not able to generate any vehicles. Also, both MSNMT models with `wait-3` and `Full` correctly captured that there were eight men, whilst both SNMT models incorrectly predicted about one and two men. From these results, we can conclude that visual clues positively impact generated translations where there is still a lack of textual information, especially when we deal with language pairs with different word order.

4.7 Summary

This chapter showed that our proposed approach of multimodal simultaneous neural machine translation takes advantage of visual information as an additional modality to improve the low-resource simultaneous neural machine translation. We showed that in a `wait-k` setting, our model significantly outperformed its text-only counterpart in situations where only a few input tokens are available to begin translation. We showed the importance of the visual information for simultaneous translation, especially in the low latency setup and for a language pair with word-order differences.

5 | Conclusions and Future Implications

In this thesis, we improved the performance of low-resource NMT by using additional information. For this purpose, we used two types of additional information with regard to text and image modalities.

In this work, different language pairs were considered, wherein each contained a different amount of available parallel data with different domains.

First, we proposed advanced pseudo-parallel corpora filtration methods that yield high-quality pseudo-parallel corpora, which further can be used to train more accurate MT systems for low-resource language pairs in Chapter 2. Our findings suggest that translation accuracy depends on both the size and quality of the training data.

Then we introduced Japanese↔Russian NMT system using out-of-domain parallel data in Chapter 3. The difficult part of this task was unknown tokens due to the difficult news domain covering various topics and extremely low-resource available parallel data. To address this issue, we investigated the coverage of translatable tokens by training M2M Transformer using an in-domain pivot parallel corpora. As a result, we found out that the proposed system can translate more tokens by taking advantage of additional pivot parallel corpora.

In the news domain, there is also a problem of completely new tokens, which is a type of unknown tokens, that cannot be dealt with by simply increasing training data coverage since new information is out every day. Therefore, we plan to tackle the problem of new tokens that cannot be introduced by using additional corpora.

We also empirically confirmed the limited success of well-established solutions when restricted to in-domain data. Then, as a result, to incorporate out-of-domain data, we proposed a multilingual multistage fine-tuning approach and observed that it substantially improves Japanese↔Russian translation compared to strong baselines.

In the future, we plan to confirm further fine-tuning for each of the specific translation directions. We will also explore the way to exploit out-of-domain pseudo-parallel data, better domain-adaptation approaches, and additional challenging language pairs.

Finally, we proposed a multimodal simultaneous neural machine translation approach that takes advantage of visual information as an additional modality to compensate for the shortage of input text information in the simultaneous neural machine translation in Chapter 4. We showed that in a `wait-k` setting proposed model significantly outperformed its text-only counterpart in situations where only a few input tokens were available to begin translation. We showed the importance of the visual information for simultaneous translation, especially in the low latency setup and for a language pair with word-order differences.

We created a separate model for each value of `wait-k`. However, in future work, we plan to experiment with having a single model for all `k` values (Zheng et al., 2019a). Furthermore, we acknowledge the importance of investigating MSNMT effects on more realistic data (e.g., TED), where the utterance does not necessarily match a shown image while speaking and/or where its context can not be guessed from the shown image.

In the future, we will combine all proposed methods to further create more robust models for low-resource language pairs. We also plan to address zero-resource languages that have almost no written resources and create a robust model for unrelated information from other modalities such as video and audio.

This work was mostly concentrated on Japanese↔Russian MT. Since I can speak both Russian and Japanese, I was able to analyze this low-resource language pair and create a new benchmark dataset for future research of such low-resource long-distance language pair. Until this work, research on Russian and Japanese languages and multimodal translation have been mainly related to Western languages. Showing that image information plays a greater role between very different languages such as Japanese and English, this work made it possible to develop multimodal, simultaneous translation for the other languages outside of mainstream languages of Europe and the United States.

Bibliography

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, 2019.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *International Conference on Learning Representations*, 2018.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, 2011.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Nicola Bertoldi and Marcello Federico. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, 2009.

Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost van de Weijer. LIUM-CVC submissions for WMT17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation*, pages 432–439, 2017a.

Ozan Caglayan, Mercedes García-Martínez, Adrien Bardet, Walid Aransa, Fethi Bougares, and Loïc Barrault. NMTPY: A flexible toolkit for advanced neural machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 109(1):15–28, 2017b.

Ozan Caglayan, Pranava Madhyastha, Lucia Specia, and Loïc Barrault. Probing the need for visual context in multimodal machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4159–4170, 2019.

Iacer Calixto, Qun Liu, and Nick Campbell. Doubly-attentive decoder for multi-modal neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1924, 2017.

Ali Cevahir and Koji Murakami. Large-scale multi-class and hierarchical product categorization for an e-commerce giant. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 525–535, 2016.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1965–1974, 2016.

Colin Cherry and George Foster. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, 2012.

Kyunghyun Cho, Bart van Merriënboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. An Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 385–391, 2017.

Trevor Cohn and Mirella Lapata. Machine Translation by Triangulation: Making Effective Use of Multi-Parallel Corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 728–735, 2007.

Ryan Cotterell and Julia Kreutzer. Explaining and generalizing back-translation through wake-sleep. *CoRR*, abs/1806.04402, 2018.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. Incremental decoding and training methods for simultaneous translation in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 493–499, 2018.

Michael Denkowski and Graham Neubig. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 18–27, 2017.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, 2018.

Desmond Elliott. Adversarial evaluation of multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2974–2978, 2018.

Desmond Elliott and Àkos Kádár. Imagination improves multimodal translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 130–141, 2017.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. Multi30k: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, 2016.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 215–233, 2017.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, 2016.

Spandana Gella, Desmond Elliott, and Frank Keller. Cross-lingual visual verb sense disambiguation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1998–2004, 2019.

Dirk Goldhahn, Maciej Sumalvico, and Uwe Quasthoff. Corpus collection for under-resourced languages with more than one million speakers. In *Workshop on Collaboration and Computing for Under-Resourced Languages (CCURL), Language Resources and Evaluation Conference*, pages 67–73, 2016.

Alvin Grissom II, He He, Jordan Boyd-Graber, John Morgan, and Hal Daumé III. Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1342–1352, 2014.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor OK Li. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1, Long Papers)*, pages 1053–1062, 2017.

Barry Haddow and Philipp Koehn. Analysing the effect of out-of-domain data on SMT systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432, 2012.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016b.

Tosho Hirasawa, Hayahide Yamagishi, Yukio Matsumura, and Mamoru Komachi. Multimodal machine translation with embedding prediction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 86–91, 2019.

Julian Hitschler, Shigehiko Schamoni, and Stefan Riezler. Multimodal pivots for image caption translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2399–2409, 2016.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, 2018.

An-Chang Hsieh, Hen-Hsen Huang, and Hsin-Hsi Chen. Uses of monolingual in-domain corpora for cross-domain adaptation with hybrid MT approaches. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation*, pages 117–122, 2013.

Kenji Imamura, Atsushi Fujita, and Eiichiro Sumita. Enhancement of encoder and attention using target monolingual corpora in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 55–63, 2018.

Aizhan Imankulova, Takayuki Sato, and Mamoru Komachi. Improving Low-resource Neural Machine Translation with Filtered Pseudo-parallel Corpus. In *Proceedings of the 4th Workshop on Asian Translation*, pages 70–78, 2017.

Aizhan Imankulova, Raj Dabre, Atsushi Fujita, and Kenji Imamura. Exploiting Out-of-Domain Parallel Data through Multilingual Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 128–139, 2019.

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 967–975, 2007.

Melvin Johnson, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017.

Tomoyuki Kajiwara, Danushka Bollegala, Yuichi Yoshida, and Ken-ichi

Kawarabayashi. An iterative approach for the global estimation of sentence similarity. *PLOS ONE*, 12(9):e0180885, 2017.

Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *The International Conference on Learning Representations*, 2015.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of Association for Computational Linguistics 2017, System Demonstrations*, pages 67–72, 2017.

Tom Kocmi and Ondřej Bojar. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, 2018.

Philipp Koehn. Europarl: A Multilingual Corpus for Evaluation of Machine Translation, 2002.

Philipp Koehn and Rebecca Knowles. Six Challenges for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, 2017.

Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003.

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. 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, 2007.

Surafel M Lakew, Quintino F. Lotito, Matteo Negri, Marco Turchi, and Marcello Federico. Improving zero-shot translation of low-resource languages. In *Proceedings of the 14th International Workshop on Spoken Language Translation*, pages 113–119, 2017.

Surafel M Lakew, Mauro Cettolo, and Marcello Federico. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 641–652, 2018.

Chiraag Lala and Lucia Specia. Multimodal lexical translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3810–3817, 2018.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*, 2018a.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, 2018b.

Jindřich Libovický and Jindřich Helcl. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–202, 2017.

Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 605–612, 2004a.

Chin-Yew Lin and Franz Josef Och. Orange: A method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 501–507, 2004b.

Minh-Thang Luong and Christopher D Manning. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation*, pages 76–79, 2015.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, 2019.

Benjamin Marie and Atsushi Fujita. Unsupervised neural machine translation initialized by unsupervised statistical machine translation. *CoRR*, abs/1810.12703, 2018.

Shigeki Matsubara, Kiyoshi Iwashima, Nobuo Kawaguchi, Katsuhiko Toyama, and Yoichi Inagaki. Simultaneous Japanese-English interpretation based on early prediction of English verb. In *Proceedings of The Fourth Symposium on Natural Language Processing*, pages 268–273, 2000.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *International Conference on Learning Representations*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pages 3111–3119, 2013b.

Robert C. Moore and Will Lewis. Intelligent Selection of Language Model Training Data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL) Short Papers*, pages 220–224, 2010.

Koji Murakami, Kiyoshi Sudo, and Keiji Shinzato. 参照文を用いない暫定的な翻訳評価と翻訳辞書作成ツールの開発. In *Proceedings of the Twenty-third Annual Meeting of the Association for Natural Language Processing*, pages 1188–1191, 2017.

Hideki Nakayama, Akihiro Tamura, and Takashi Ninomiya. A visually-grounded parallel corpus with phrase-to-region linking. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4204–4210, 2020.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. ASPEC: Asian Scientific Paper Excerpt Corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 2204–2208, 2016.

Xing Niu, Michael Denkowski, and Marine Carpuat. Bi-directional neural machine translation with synthetic parallel data. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 84–91, 2018.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Syntax-based simultaneous translation through prediction of unseen syntactic constituents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 198–207, 2015.

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

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision*, 123(1):74–93, 2017.

Holger Schwenk. Investigations on large-scale lightly-supervised training for statistical machine translation. In *Proceedings of International Workshop on Spoken Language Translation*, pages 182–189, 2008.

Kilian G Seeber. Simultaneous interpreting. In *The Routledge Handbook of Interpreting*, pages 91–107. 2015.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, August 2016a.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, 2016b.

Yangqiu Song and Dan Roth. Unsupervised sparse vector densification for short text similarity. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1275–1280, 2015.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation: (Volume 2: Shared Task Papers)*, pages 543–553, 2016.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 3104–3112, 2014.

Christoph Tillmann and Jian-ming Xu. A simple sentence-level extraction algorithm for comparable data. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 93–96, 2009.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 25–32, 2007.

Masao Utiyama and Hitoshi Isahara. Reliable Measures for Aligning Japanese-English News Articles and Sentences. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 72–79, 2003.

Masao Utiyama and Hitoshi Isahara. A comparison of pivot methods for phrase-based statistical machine translation. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 484–491, 2007.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of 30th Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

Longyue Wang, Derek F Wong, Lidia S Chao, Yi Lu, and Junwen Xing. A Systematic Comparison of Data Selection Criteria for SMT Domain Adaptation. *The Scientific World Journal*, 2014:1–10, 2014.

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, Å□ukasz 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, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

Eray Yıldız, Ahmed Cüneyd Tantuğ, and Banu Diri. The effect of parallel corpus quality vs size in English-to-Turkish SMT. In *Proceedings of the Sixth International Conference on Web services and Semantic Technology*, pages 21–30, 2014.

Jiajun Zhang and Chengqing Zong. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, 2016.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. Joint training for neural machine translation models with monolingual data. In *AAAI Conference on Artificial Intelligence*, pages 555–562, 2018.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. Simultaneous translation with flexible policy via restricted imitation learning. pages 5816–5822, 2019a.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. Simpler and faster learning of adaptive policies for simultaneous 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 1349–1354, 2019b.

Renjie Zheng, Mingbo Ma, Baigong Zheng, and Liang Huang. Speculative beam search for simultaneous 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 1395–1402, 2019c.

Renjie Zheng, Mingbo Ma, Baigong Zheng, Kaibo Liu, and Liang Huang. Opportunistic decoding with timely correction for simultaneous translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 437–442, 2020.

Mingyang Zhou, Runxiang Cheng, Yong Jae Lee, and Zhou Yu. A visual attention grounding neural model for multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3643–3653, 2018.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, 2016.

Publications

Journal Papers

1. Aizhan Imankulova, Takayuki Sato, Mamoru Komachi. **Filtered Pseudo-Parallel Corpus Improves Low-Resource Neural Machine Translation**. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP). 19, 2, Article 24 (October 2019), 16 pages.

International Conference Papers

1. Siti Oryza Khairunnisa, Aizhan Imankulova and Mamoru Komachi. **Towards a Standardized Dataset on Indonesian Named Entity Recognition**. In The 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing 2020 Student Research Workshop (AACL-IJCNLP SRW), pp.64‑71 December, 2020.

2. Ikumi Yamashita, Satoru Katsumata, Masahiro Kaneko, Aizhan Imankulova and Mamoru Komachi. **Cross-lingual Transfer Learning for Grammatical Error Correction**. The 28th International Conference on Computational Linguistics (COLING), pp.4704-4715. December, 2020.

3. Aizhan Imankulova, Masahiro Kaneko, Tosho Hirasawa and Mamoru Komachi. **Towards Multimodal Simultaneous Neural Machine Translation**. In Proceedings of the Fifth Conference on Machine Translation (WMT), pp.540–549. Online. November, 2020.

4. Masahiro Kaneko, Aizhan Imankulova, Tosho Hirasawa and Mamoru Komachi. **English-to-Japanese Diverse Translation by Combining Forward and Backward Outputs**. The 4th Workshop on Neural Generation and Translation (WNGT): Simultaneous Translation And Paraphrase for Language Education (STAPLE) English-to-Japanese track, pp.134-138. 2020.

5. Aizhan Imankulova, Masahiro Kaneko and Mamoru Komachi. **Japanese-Russian TMU Neural Machine Translation System using Multilingual**

Model for WAT 2019. In Proceedings of the 6th Workshop on Asian Translation (WAT2019): News Commentary task, pp.165-170. Hong Kong, China. November 4, 2019.

6. Aizhan Imankulova, Raj Dabre, Atsushi Fujita, and Kenji Imamura. **Exploiting Out-of-Domain Parallel Data through Multilingual Transfer Learning for Low-Resource Neural Machine Translation**. In Proceedings of Machine Translation Summit XVII Volume 1: Research Track (MT Summit), pp.128-139. Dublin, Ireland. August, 2019.

7. Aizhan Imankulova, Takayuki Sato and Mamoru Komachi. **Improving Low-Resource Neural Machine Translation with Filtered Pseudo-Parallel Corpus**. In Proceedings of the 4th Workshop on Asian Translation (WAT2017), pp.70-78. Taipei, Taiwan. November 27, 2017.

Domestic Conference Presentations

1. 山下郁海, 勝又智, 金子正弘, Imankulova Aizhan, 小町守. 言語間での転移学習を用いたロシア語文法誤り訂正. 言語処理学会第26回年次大会, pp.1324-1327. March 19, 2020

2. Imankulova Aizhan, 金子正弘, 平澤寅庄, 小町守. 画像を使用したマルチモーダルニューラル同時翻訳. NLP若手の会第14回シンポジウム. August 27, 2019.

3. 山下郁海, 勝又智, 金子正弘, Imankulova Aizhan, 小町守. 英語からロシア語への転移学習を用いた文法誤り訂正. NLP若手の会第14回シンポジウム. August 27, 2019.

4. 金子正弘, Imankulova Aizhan, 小町守. 転移学習を用いてコンテキストを考慮した系列変換タスクにおけるリランキング. NLP若手の会第13回シンポジウム. August 28, 2018.

5. Aizhan Imankulova, Koji Murakami. **Preliminary Experiments toward NMT on E-commerce Product Titles**. 言語処理学会第24回年次大会, pp.893-896. March 14, 2018.

6. Imankulova Aizhan, 佐藤貴之, 小町守. 逆翻訳による高品質な大規模

擬似対訳コーパスの作成. 言語処理学会第23回年次大会. March 14, 2017.