Approaches for Building an Effective System of Multi-domain Neural Machine Translation

A Thesis Submitted for the Degree of Doctor of Philosophy,
Studies in Language and Culture,
Graduate School of Language and Culture,
Osaka University

by

LUO, Wentao
罗　　雯涛

March　2020

Acknowledgments

First of all, I would like to sincerely thank my supervisors.

I would like to express my sincere gratitude to Prof. Hisashi Iwane for his kind guidance. Under his supervision, I was encouraged to challenge the research topics and to overcome difficulties during study life in Osaka University.

I am grateful acknowledge Assoc. Prof. Bor Hodošček for providing valuable comments on my research and helping me greatly in writing this thesis.

I sincerely thank Prof. Tomoji Tabata for his detailed review and valuable comments.

I would like to thank Prof. Youichi Miyamoto for his comment about this thesis.

I also deeply appreciate Prof. Yoshihiko Hayashi for giving a great variety of comment about natural language processing.

Secondly, I would like to thank Kodensha for providing a good opportunity for me to engage in research related to neural network translation. At the same time, I would like to thank my colleagues in Kodensha for providing ideas for my research. Some of my basic research work is also inseparable from their assistance. In addition, my colleagues have given me great support in all aspects other than work.

Finally, I thank my family members for giving me unlimited support and endless trust.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Machine translation, often abbreviated as MT, has been one of the major challenges in natural language processing (NLP). The development of high-quality MT systems and services has become increasingly important as the amount of information/knowledge exchanged by people has been rising exponentially, given the rapid progress of communication technologies. To fulfill the social demands for cross-lingual information exchange, a broad range of NLP technologies have been developed so far, and these MT-originated technologies have also enhanced the development of other types of NLP-related systems and services, such as information retrieval, information extraction, and text summarization.

The development of MT technologies has its own rich history, as briefly reviewed in the succeeding section. Of particular prominence is neural machine translation (NMT), as it enables far better translation quality compared with MT systems/services that rely on traditional frameworks such as rule-based or statistical translation. One of the most popular and widely-available examples of the NMT system is Google Translate[1], which currently provides services in more than one hundred languages. While the present state of neural machine translation achieves a reasonable degree of efficacy, its translation quality still does not match human experts. Among many issues to be addressed in an effort to achieve a higher level of quality is how to make the system accommodate to sentences or expres-

[1] https://translate.google.co.jp/

sions specific to an understudied domain. A domain is a category of text classified according to its content, form, and function. A typical NMT system that is trained using a general domain corpus is capable of handling general domain texts with a practical accuracy. It may, on the other hand, struggle to deal with sentences specific to a particular domain for which the system has no prior optimization.

To improve translation quality, NMT needs to draw on a training corpus with an extensive domain coverage. However, it is not realistic, for logistical reasons, to consider training the system with a corpus that can encompass all conceivable domains. A possible solution would be to optimize a NMT system for a specific target domain. However, there is one notable disadvantage to this domain-specific translation approach. The translation system is assumed to be adapted to a particular domain, based on a choice users make in advance. This would not work efficiently when users are not fully aware of the inventory of domains available or when they do not make an appropriate selection of domains to suit their need.

To address the issue of domain adaptability and to achieve a better quality of translation across multiple domains, we, in this thesis, propose an architecture and training schemes for NMT systems that facilitates multiple-domain translation. The essential components of the proposed system are techniques for facilitating appropriate domain selection and organizing domain ensemble. The former technique enables a proper selection of relevant translation domain models from among pre-trained domain models, whereas the latter technique optimally decodes a sequence of words in the target language by adequately combining the predictions made by the selected domain models.

In the rest of this chapter, before introducing neural machine translation in detail, we will briefly review the history of machine translation in Section 1.1. Then, we will introduce NMT to discuss possible problems NMT is presently confronted with in Section 1.2. In Section 1.3, we will discuss multi-domain translation, a scenario of machine translation, and describe our motivation for building an effective NMT system for multi-domain translation. Finally, we will

describe the organization of this thesis in Section 1.4.

Remind that throughout this thesis, following the convention of the area of machine translation, the language of the input and output text are respectively referred to as source language and target language. For instance, when translating English text to Chinese text. English is the source language. Chinese is the target language.

1.1 Brief History of Machine Translation

Research on machine translation began in the 1950s with most of efforts dedicated to translating texts from Russian into English. The enterprise faced harsh criticism in 1964, when a report by the American Automatic Language Processing Advisory Committee (ALPAC, 1966) pointed out that machine translation research was impractical given the fact that various studies of machine translation had not achieved expected results. The report called for a need for fundamental research with suggestion for concentrating greater resources in support of human translation. After experiencing a short period of diminished funding in the 1960s, the research community regained its momentum in the 1970s, ushering in a boom in the late 1990s.

One of the prominent reasons for the recovery is partly due to the acceleration of globalization and the rapid development of the internet. Traditional human translation has been unable to meet people's growing needs. In addition, with the steady development of computer technologies, a massive amount of texts have been produced and stored in digital formats, making it possible to adopt data-driven NLP approaches.

Types of machine translation systems before the 2010s can be roughly divided into Rule-based Machine Translation (RBMT), Example-based Machine Translation (EBMT), and Statistical Machine Translation (SMT), all of which in turn are explained in detail. Neural Machine Translation (NMT), which is the focus of this thesis, sprouted up in 2014. The technique was based on a sequence-to-sequence

transformation process realized by the recurrent neural network (RNN) architecture. Neural network-based methods have rapidly taken over the mainstream of MT technology. Just two years after the invention, Google replaced its SMT-based engine of the Web-based public translation service with the new NMT-based engine in the autumn of 2016. NMT has since been dominant not only in the development of translation systems/services but also in research on MT technologies.

1.1.1 Rule-based Machine Translation

Since machine translation is a task that converts a word sequence of a source language into a corresponding word sequence in a target language[2], the research community have tried to devise an effective set of translation rules for performing an accurate translation. Based on a set of transfer translation rules on different language levels, Rule-based Machine Translation (RBMT) can be divided into three sub-types: Direct translation, Syntactic and Semantic transfer, and Interlingua-based translation, which is not a transfer approach in a strict sense. Figure 1.1 illustrates a transfer processes on several language levels, which is often referred to as a "machine translation pyramid" or "translation triangle."

1. Direct translation

   The first sub-type of RBMT is direct translation. Basically, an MT system based on the direct translation method performs a word-by-word translation usually by looking up a bilingual dictionary.

2. Syntactic and Semantic Transfer

   The direct translation method can only be applied to a pair of languages in which the word order is essentially the same, meaning that it cannot be applied to pairs of languages that have different word-orders. In an effort to overcome this serious limitation, syntactic-level transfer methods have been developed. This method first syntactically scans a source text and analyze its syntactic structure. It then transforms the syntactic structure in

[2]See the previous footnote about source language.

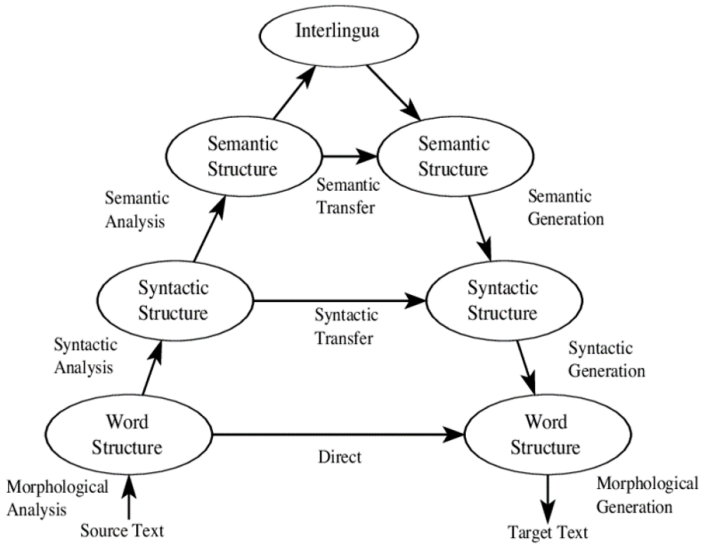


Figure 1.1: Machine translation pyramid
(adopted from Dorr et al. (2002), figure 3)

the source language into the corresponding syntactic structure in the target language. In this processing stage, a set of hand-crafted transfer rules is applied at a syntactic level. Finally, this method generates a sequence of words in the target language from the source syntactic structure. These three processes are respectively referred to as Analysis, Transfer, and Generation, as illustrated in Figure 1.1. Similar to syntactic transfer, semantic transfer performs a source-to-target conversion process at a semantic level, which generally is in predicate-argument structures.

3. Interlingua-based translation

   The interlingua-based translation method first produces a language-independent structure for an input text string by applying a set of source language-dependent analysis rules at morphological, syntactic, and semantic layers. It then generates a word sequence in the target language by exploiting a set of target language-dependent generation rules. That is, the entire translation process is made up of two parts: Analysis and Generation, meaning that

there is no Transfer process. This is why we remarked that this method was not a transfer method in a strict sense in the preceding paragraph.

The interlingua-based translation method obviously has its strength in developing a multilingual translation system, where many source and target languages have to be translatable. That is, only analysis and generation modules are necessary for each language, as no language pair-wise transfer modules are necessary.

However, defining and designing expressions in an interlingua is hard to accomplish: it has to be *lingua franca*, that is, it should be language-independent and universal. Some researchers have tried to apply a logic-based language to serve as a foundation for interlingua representations, but it has turned out that there are several innate drawbacks, including a lack of expressiveness in conveying fine-grained semantic nuances. The methods that are purely based on interlingua representations now rarely attract the attention of MT/NLP researchers, even if they are, in a sense, idealistic.

Rule-Based Machine Translation (RBMT) subsumes translation methods inspired by the "machine translation pyramid" in the strict sense. Mainstream rule-based machine translation systems are all inspired by Chomsky's transformational generative grammar, which tries to deduce infinitely long text strings using a limited set of rules.

A major advantage of RBMT is its flexibility in building a set of rules: fine-grained rules and coarse-grained rules can co-exist in the rule set, allowing its developers to add rules with arbitrary granularities. This feature actually cuts two ways: an arbitrary addition or modification of rules can result in tipping the balance in one direction or the other, causing conflicting rules.

The advantages and disadvantages of RBMT (Vauquois & Boitet, 1985) are summarized as follows.

Pros:

+ Coarse-grained rules can cope with general linguistic phenomena.

+ Fine-grained rules can deal with idiosyncratic phenomena.

Cons:

– Translation rules have to be compiled manually, which can entail a huge implementation cost.
– Translation rules can be highly subjective, making it difficult to maintain consistency among the rules developed by different authors.
– Hand-crafted translation rules cannot exhaust all idiosyncratic expressions.

1.1.2 Example-based Machine Translation

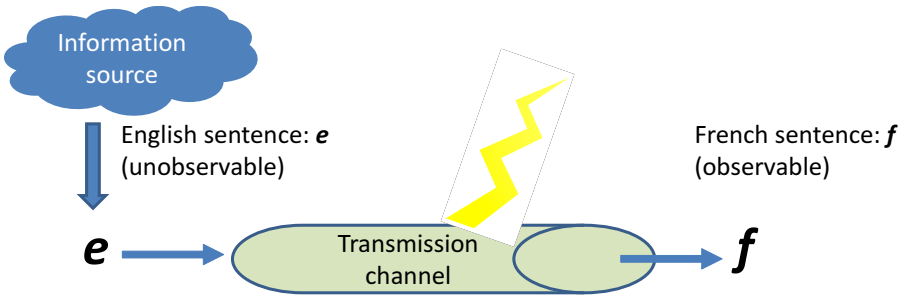
As discussed in the previous section, RBMT has a serious drawback in maintaining coherence between hand-crafted translation rules. Example-Based Machine Translation (EBMT), first proposed by Nagao (1981), is one solution to addressing this drawback. The idea of EBMT is inspired by the notion of "translation by analogy": when translating a new sentence, human translators tend to look for similar cases in an archive of already translated sentences, also known as as a translation bank. As a translation bank is basically just a collection of individual translation examples, EBMT is free from the issue of rule consistency, which is a crucial issue in RBMT. Therefore, an EBMT system first locates a sentence similar to the current input sentence in its translation bank, and then alters the corresponding target sentence to match the current input.

For example, when an EBMT system translates an input sentence, "I am going to the cinema". Suppose the system find a similar sentence "I am going to the theater" in its translation bank. The system then alters the corresponding translation to generate a translation tailored to the current input. More specifically, in this case, the translation word for "cinema" is simply replaced by the translation word of "theater." The greater number of examples the translation bank incorporates, the better translation outputs can be obtained. Of further merit, the EBMT approach is suitable for building a translation system between two widely different languages in terms of lexis, syntax, and semantics: English and Japanese,

for instance. By applying the translation by analogy approach, no deep linguistic analysis is required in principle. This approach provided insight to the MT research community, leading to the development of statistical machine translation (SMT), which will be discussed in the next section.

1.1.3 Statistical Machine Translation

Statistical Machine Translation (SMT) is a mathematical translation approach. More specifically, it is inspired by the "noisy channel model" (Brill & Moore, 2000), which was originally developed as an outgrowth of mathematical information theory.



Translation is a process for recovering the original message *e*, given the observable message *f*.

$$e = \arg\max_{e'} p(e' \mid f)$$

Figure 1.2: Noisy channel model

As illustrated in Figure 1.2, the model assumes that an input message (sentence) e is transmitted through a transmission channel that is often affected by noise. The transmitted message f, being more or less disturbed on the way, has to be decoded to recover the original message e at the receiving point. With this analogy, the received message and the original message correspond to the source sentence and the target sentence in translation, respectively. In sum, this translation model is considered as a process of recovering the unobservable original message e, given

the observable received message f. We must remind here that the source sentence corresponds to the output from the communication channel and that the target sentence corresponds to the input to the communication channel.

To recover the original signal, a statistical decoding algorithm is applied. Such a decoding algorithm is generally referred to as a statistical translation model.

The fundamental mathematical formula can be given as follows.

$$\underset{e'}{argmax}\, p(e'|f) = \underset{e'}{argmax}\, \frac{p(f|e')p(e')}{p(f)} \tag{1.1}$$

That is, the translated sentence e for the input sentence f is chosen from the candidate translation sentences e's that gives the maximum conditional probability $p(e'|f)$. This conditional probability can be computed by the right hand side of the equation, where the numerator is essential, as the denominator can be independent of e'. As the first term $p(f|e')$ captures the degree of correspondence between f and e', it is often referred to as a translation model. On the other hand, the second term $p(e')$ measures the likelihood of a translated candidate e' in the target language, hence referred to as a language model. To build this model, a corpus of source-target paired sentences (bilingual corpus) is necessary. By using a bilingual corpus, a language model of the target language is built together with a translation model that captures the correspondences between the source and target languages.

An array of feasible SMT models were first proposed in 1993 by a research team at IBM Brown et al. (1993). Their first model, called IBM Model-1, computes $p(e'|f)$ by relying on a simple word-level alignment method. The succeeding models (IBM Model-2 through Model-5) considered more complex cases. These models, offering better translation performances, posed a problem of computational complexity: models with greater complexity take a longer training time.

Furthermore, the unit of alignment operation has basically remained at the word-level, which often affected translation quality. To properly translate certain types of expressions, alignment operations have to be performed by grouping a sequence of words as a processing unit rather than a single word.

To overcome known limitations, statistical models have evolved continuously.

The most significant step in the evolution is the adoption of a word sequence (phrase[3]) as a unit of alignment. As a sequence of words can capture more fine-grained and useful cross-lingual correspondences than a single word, these Phrase-based Statistical Machine Translation (PBMT) models (Koehn et al., 2003) achieved a higher translation accuracy.

In the same year, Och (2003) proposed Minimum Error Rate Training, enabling the optimization of a translation model by maximizing translation quality metrics. In addition, as their method significantly reduced the computational cost, SMT systems were commercialized. Under their impetus, the PBMT model had become a de facto standard of the MT system. Further enhancements were made to improve the phrase-based model by the work of Chiang (2005) and Chiang (2007), which proposed a hierarchical phrase model. In their hierarchical phrase-based model, the proposed algorithm deals with input phrases at different linguistic levels to ensure a more accurate output. During the decade from 2006 to 2016, PBMT was studied intensively as the mainstream machine translation model.

1.2 Neural Machine Translation

In recent years, with the rapid development of deep learning, Neural Machine Translation (NMT) has made significant progress. In many languages, the performance of neural network machine translation has been dramatically improved and far exceeded the traditional mainstream machine translation technologies. Currently, NMT research is a hot topic in natural language processing community.

At the same time, major companies in the industry have also invested workforce and material resources in developing their own neural network machine translation systems. NMT has become the core technology of major machine translation systems. It triggered development of translation tools. The translation tools have become increasingly common and an essential part of people's lives. As a result of various studies of NMT, there is an extensive proliferation of toolkits available

[3]It must be mentioned that the term *phrase* in this context does not necessarily mean a syntactic phrase. It just refers to a sequence of words that is statistically prominent in the training corpus.

for research, development, and deployment of neural machine translation systems: Nematus[4], Marian[5], OpenNMT[6], xnmt[7], Sockeye[8], T2T[9], and FairSeq[10], among others.

In this section, we will introduce the basic framework of NMT, the SOTA (state of the art) framework of NMT, and challenges for NMT.

1.2.1 Basic Framework of NMT

A basic framework of NMT was based on a recurrent neural network (RNN) model. For more detail, we will briefly explain three parts of the basic framework of NMT, as shown in Figure 1.3: encoder, decoder, and attention layer.

1. Encoder

   The encoder models the input sentence with a bidirectional recurrent neural network. NMT maps the source language vocabulary into a sequence of word vectors to obtain a sequence of encoded input vectors $(x_1, \cdots, x_n)$. The number n is the length of the source language sentence. The encoder encodes the input sequence as a vector of hidden layers $(h_1. \ \cdots .h_n)$. Forward and backward RNNs (bidirectional RNN) both update the hidden layer vector of each word.

2. Decoder

   The decoder predicts the word y using the target language vocabulary through the attention layer and hidden layers s.

   NMT concatenates the hidden state layer s and attention layer to obtain the hidden layer $s^{\sim}$. The output word y is calculated by performing a softmax

[4]Nematus: https://github.com/EdinburghNLP/nematus
[5]Marian: https://marian-nmt.github.io/
[6]OpenNMT: http://opennmt.net/
[7]xnmt: https://github.com/neulab/xnmt
[8]Sockeye: https://github.com/awslabs/sockeye
[9]T2T: https://github.com/tensorflow/tensor2tensor
[10]FairSeq:https://github.com/pytorch/fairseq

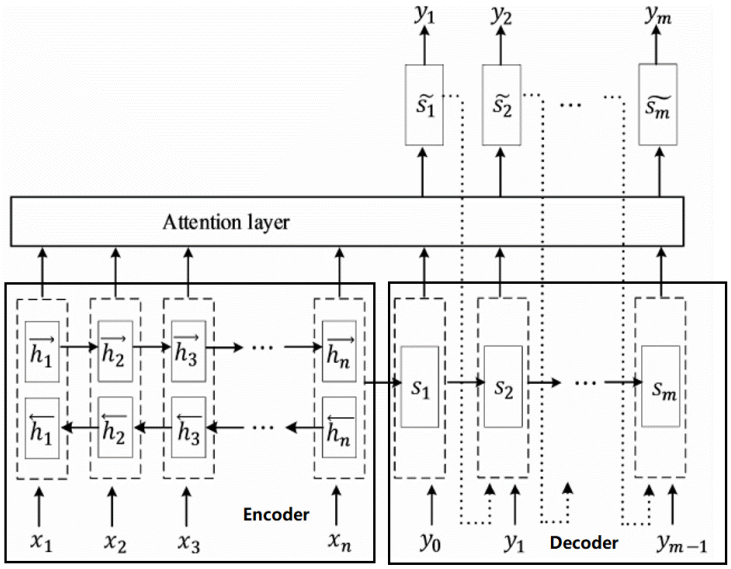


Figure 1.3: Basic framework of NMT

operation[11] on probability distribution $s^{\sim}$. In short, the encoder accepts the input text (word x) of the source language, then the decoder accepts the encoder's information, and finally gives the translation (word y), state by state (from state s_1 to state s_m). Its main feature is that during decoding, each step dynamically uses the attention mechanism to capture the source language context information related to the current query.

3. Mechanism of Attention layer
   If the decoder only uses the state (h) outut by the encoder at the last moment (h_n), this may cause two problems.

   (a) The last hidden state of the encoder has a higher correlation with the words at the end of the sentence, making it challenging to retain the information at the beginning of the sentence.

[11]When predicting an output word, the NMT system calculates a probability vector. Each element in this vector corresponds to the probability of each word in the vocabulary. At that time, the operation that takes MAX is a differentiable form that can be learned by backpropagation. Hence, the computational cost is directly proportional to the size of the vocabulary.

(b) The encoder accepts input text in order and produces the hidden state containing word order information. Therefore, The translation of the decoder is also very likely to be performed in the order of the input sentence. When translating sentences between two languages with different word orders, such as between English and Japanese, we need to break the word order of the input sentence while translating.

In order to solve the above problems, researchers began to consider using the attention mechanism to obtain the context information in the input sentence. The attention layer (as shown in Figure 1.3) is added into the encoder-decoder to preserve the context information of the input sentence produced by the encoder when the decoder processes.

There are many specific attention mechanisms. As an example, Luong et al. (2015) proposed global attention and local attention.

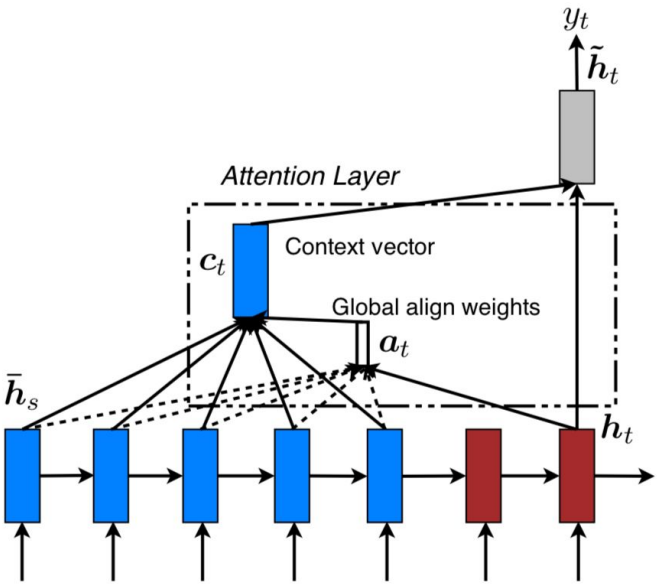


Figure 1.4: Global attention
(adopted from Luong et al. (2015), figure 2)

As shown in Figure 1.4, when global attention calculates the context infor-

mation (context vector) for h_t, it computes the probability of all the hidden state ($\bar{h}_s$) generated by the encoder.

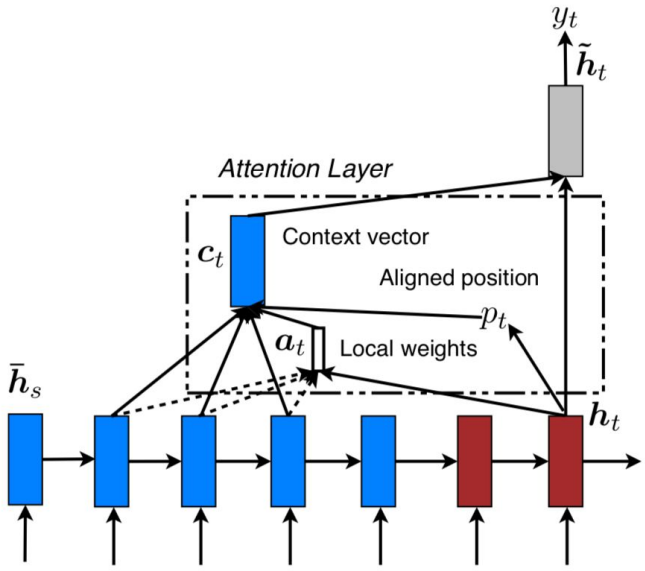


Figure 1.5: Local attention
(adopted from Luong et al. (2015), figure 3)

At the same time, because global attention scans all source hidden states ($\bar{h}_s$) each time, this raises a problem of high computational cost. In order to improve efficiency when translating long sentences, local attention only focused on a small window of the hidden state ($\bar{h}_s$) generated by the encoder to calculate the context vector. The window size[12] of the hidden state is based on a position-related measure (p_t in Figure 1.5).

1.2.2 Advanced Framework of NMT: Transformer

Transformer, a framework based on a self-attention mechanism, improved the basic Framework of NMT. Vaswani et al. (2017) pointed out that basic the Framework

[12]Window size determines how many hidden states need to be fetched while calculating the context vector.

of NMT combining RNN (encoder and decoder) and attention mechanism has achieved good results, but because of the characteristics of RNN, it is not conducive to training data in parallel, so the model training time is often longer. Therefore, they further abandoned the traditional encoder-decoder model and proposed a framework to reinforce the utilization of the attention mechanism: Transformer. Instead of global attention and local attention, Transformer utilizes a self-attention mechanism.

In the basic NMT framework using RNN, the NMT model computes parameters in the input order in the neural network. For long-distance words, it becomes difficult to capture the characteristics between them. However, self-attention directly connects the relationship between any two words in a sentence through one calculation step, so self-attention can more effectively capture the characteristics of words at a longer distance. Additionally, in order to keep the information of word position (word order) in the sentence, Transformer also introduced a word position encoding. [13]

On the other hand, self-attention also directly helps to increase the parallelism of the calculation.

1.2.3 Challenges for NMT

Although neural network machine translation has made significant progress over traditional machine translation, there are still many issues that neural network machine translation needs to resolve. As mentioned in (Koehn & Knowles, 2017) and (Goto & Tanaka, 2017), these are four challenges facing Neural Machine Translation: domain mismatch, corpus data size, rare words, and untranslated content. As a solution to the problem of domain mismatch, this thesis will examine the topic of multi-domain translation and propose an effective system framework.

1. Domain mismatch

[13] Word position encoding adds sin variables at even positions of the word vector of each word, and cos variables at odd positions. This is explained in more detail in (Vaswani et al., 2017)

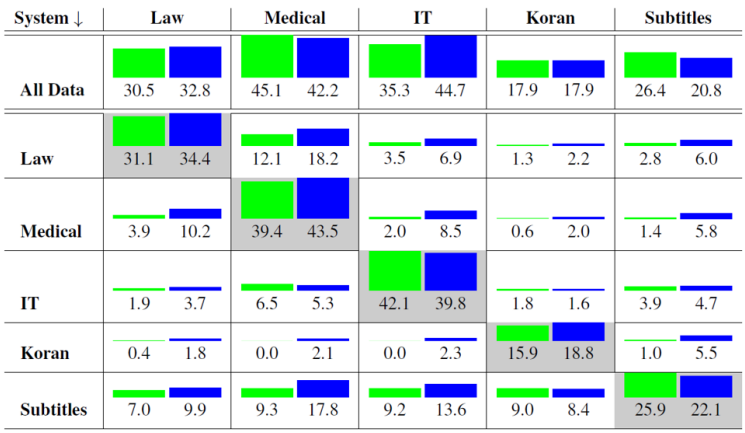


Figure 1.6: Domain mismatch
(adopted from Koehn & Knowles (2017), figure 1)

Koehn & Knowles described the situation of domain mismatch. They used corpora from different domains to train systems (systems correspond to the columns of Figure 1.6). Then they evaluate the quality of each system on each test set (test sets correspond to the rows of Figure 1.6). As shown in Figure 1.6, green represents the quality of NMT, and blue represents the quality of SMT. The evaluation value is the BLEU score, a metric that will be detailed in Chapter 7. In the case where the domain matches, the BLEU score of NMT and SMT are similar. For example, when the system (LAW) translated the test set (LAW), the BLEU score of NMT was 31.1, and the BLEU score of SMT was 34.4. However, in the case where the domain is mismatched, the quality of NMT systems was worse than SMT systems. For example, the system trained on the medical training set had a significant gap in the BLEU score as compared to the law test set (3.9 vs. 10.2).

2. Corpus data size

Another challenge is that corpus size affects the quality of NMT system. Figure 1.7 describes the learning curve concerning corpus size and transla-

tion performance of systems (in terms of BLEU). By comparing the learning curves of the SMT systems (Phrase-based and Phrase-based with Big LM) and the NMT system (Neural), we find that the learning curve of NMT is steeper in comparison with SMT. The comparison makes it evident that NMT is less effective than SMT when the corpus is small.

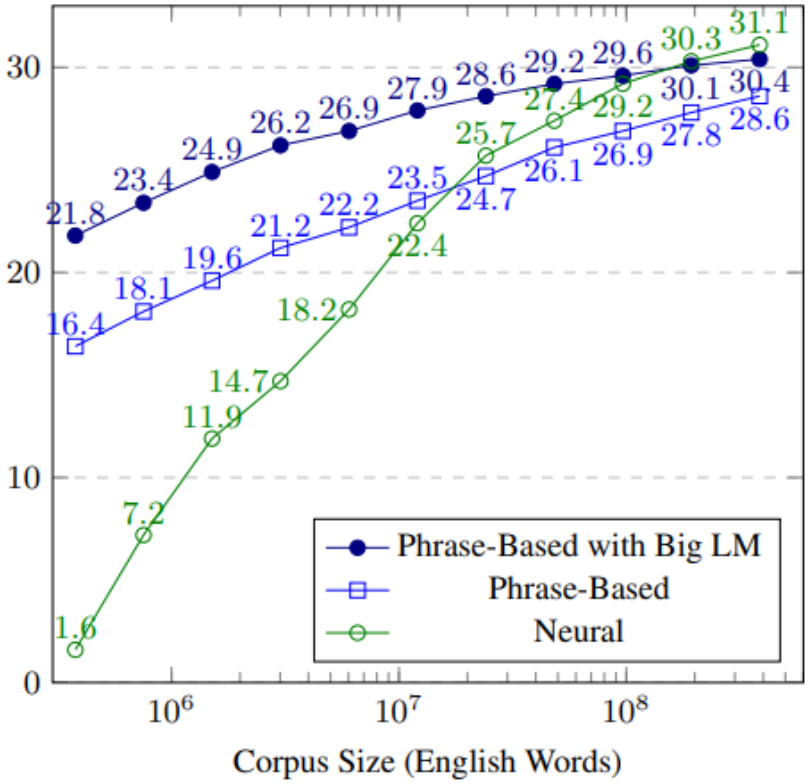


Figure 1.7: BLEU scores with varying corpus size
(adopted from Koehn & Knowles (2017), figure 3)

3. Rare words

   As mentioned earlier, since the neural network performs a softmax operation on the probability distribution of the final output word, the computational cost of this operation is expensive. In actual calculations, the vocabulary size that the NMT model can predict is limited to an order of magnitude

around 20,000 to 80,000. Therefore, the NMT model has been proven to work poorly with rare words (or low-frequency words).

4. Untranslated content

   One disadvantage of NMT compared with SMT is untranslated content. Some content is lost during the translation process from the source text to the target text. The phenomenon of untranslated content is due to a difference in the way that NMT and SMT end the translation process. SMT does not terminate the translation process until it finds a suitable translation for each word in the source text. On the other hand, NMT outputs a word sequence in turn. Once the terminator symbol is generated, NMT can terminate the translation. NMT then ignores the rest parts of the source text. In response to this problem of NMT, (Goto & Tanaka, 2017) proposed a method to detect untranslated content in NMT.

1.3 Multi-domain Translation

Let us think about the specific translation scenario of translating a text from a specific domain. If there is not enough data that matches the domain when training the model, it will cause a domain mismatch between the domain of this text and the training corpus the model used. As the challenge for NMT mentioned in Section 1.2.3, this will lead to the translation quality deteriorating.

Because NMT is quite sensitive to training corpora, each domain has its language style, sentence structure, and technical terms. For example, the word "worm" in English could refer to "insect larva" in the biology domain. In the IT domain, it could mean a virus that infects the computer program and operating system. If the NMT model trained by the corpus in the biology domain is used to translate the sentences in the IT domain, the translation result will thus be unsatisfactory.

1.3.1 Domain-specific translation

In order to match the domain, a new NMT system is often developed for a specific domain in practical applications.

For example, we can specifically train an IT domain-specific translation system for the IT domain. Then the next problem is insufficient IT data. A standard solution is to use domain adaptation. Domain adaptation is a typical methodology of transfer learning, which will be described in Chapter 3. The idea of domain adaptation is to make good use of the information of non-IT domains, which may be more readily available, to improve the translation on the IT domain.

However, the domain-specific translation produced in this way also encounters another problem, which is that the application domain of the system is too narrow. Once the input text does not belong to the domains of the training data, it will significantly reduce the translation effectiveness.

1.3.2 From Domain-specific to Multi-domain Scenario

To overcome the problem that the system is only suitable for a specific narrow domain, researchers construct a neural machine translation system in a multi-domain scenario. In this scenario, the system can accept input sentences from multiple domains. A typical multi-domain translation process is shown in Figure 1.8.

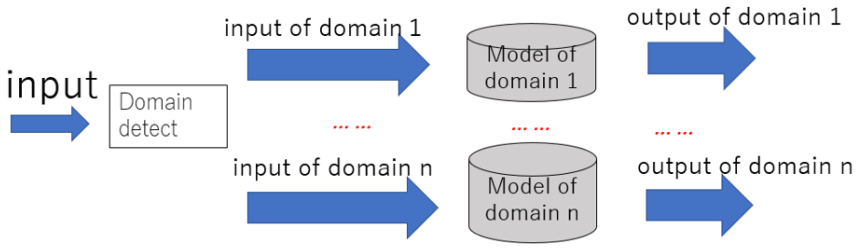


Figure 1.8: Typical multi-domain translation

This system integrates many domain-specific translation models. Whenever there is an input, the system first performs domain detection, and according to the

determined domain, submits the input to a specific domain-specific model, and finally obtains a translation.

Therefore, this system can cope with the multi-domain scenario. However, because the system needs to perform domain detection, in practical applications, the result of domain detection will primarily affect the quality of the translation output by the translation system.

As a compromise solution to improve translation accuracy, researchers typically use corpora of all available domains. They are thus able to train a system that mixes multiple domains. This system can improve the translation of available domains after sophisticated training. However, the problem with this solution is that whenever there is a translation requirement for a new domain, there is a need to retrain the model, which incurs a vast computational overhead.

Therefore, finding a better way to solve the above problems will become a crucial point of multi-domain translation, and also the focus of this thesis.

1.3.3 Motivation and Challenges

The motivation of this thesis is to build a multi-domain neural machine translation system based on model selection and model ensemble ideas. Comparing with the typical multi-domain translation systems, the multi-domain neural machine systems proposed in Chapter 2 tries to solve the following two problems encountered in practical situations.

1. If an input text does not contain domain information, is there any way to deal with it?
   For this problem, the approach based on model selection chooses the models which are most related to the input text to reduce translation difficulty. Using model selection, we can select the most related models for input text. However, there may still be substantial differences between the selected models.

2. Given that the multi-domain translation system contains multiple models,

the typical multi-domain translation system only uses one model at a time. How could we use these multiple models more effectively at the same time?

For this problem, the approach based on a model ensemble can leverage multiple translation models at once.

There are also two challenges to building a multi-domain translation system.

1. Building an appropriate specific-domain translation model.
   For specific input text, the performance of such a model could be better than a general translation model. As mentioned in the previous section, if some domains do not have enough data to train a model, it will cause the trained translation model to perform poorly on such domains.

   In the process of constructing the multi-domain translation system, we can use domain adaptation to obtain the corresponding domain-specific translation models and effectively use these models.

2. Reducing the cost of pre-processing the domain information of the input text by the user.
   For each independent domain-specific translation system, users often need to pre-process in advance to decide which system to use to translate the input text. Therefore, in order to ensure that the pre-processing is correct, the user expends considerable time and cost.

   In the multi-domain translation system, this pre-processing will be completed by the system itself, which can significantly alleviate the costs of users.

1.4 Organization of the Thesis

The organization of this thesis is shown in Figure 1.9. The chapters of this thesis will be arranged as follows.

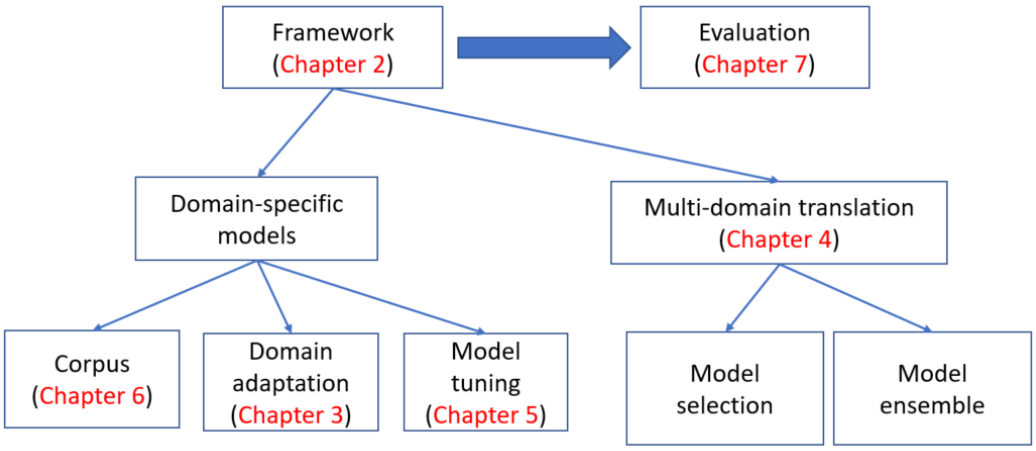


Figure 1.9: Organization of the thesis

Chapter 2 proposes the framework of the multi-domain translation system in this thesis. We discuss the key steps and approaches to achieve this system.

In Chapter 3, we explain how to train the actual domain-specific translation model. We utilize five domain adaptation methods to train the domain-specific translation model.

Chapter 4 describes approaches for text translation using the domain-specific translation models after domain adaptation. For model selection, we will use a measure to select the most suitable model for input text. The measure is inspired by a perplexity-based approach, which checks the ability of a model to translate input text. Then we ensemble the selected multiple models in the decoding phase.

In Chapter 5, as post-processing after training the domain-specific model, we adopted some tuning methods to improve the effectiveness of the domain-specific translation model. These tuning methods correspond to the various problems discovered from translation, including adding noise to the corpus data to improve the translation performance. As another tuning method, we will introduce a method of replacing words and phrases, which can effectively solve the unknown word problem.

In Chapter 6, we will discuss how to collect large-scale parallel corpora for model training. To deal with the lack of parallel corpora, we employed existing translation systems to expand the corpus.

In Chapter 7, we will define the evaluation task and the test sets. Then we will evaluate and analyze the multi-domain translation system. Thus, the effectiveness of the multi-domain translation system proposed in this thesis will be confirmed. Through the analysis of the results, we discuss some useful insights and experiences.

In Chapter 8, we summarize the work and contributions of this thesis. For multi-domain translation, this thesis proposes approaches based on model selection and model ensemble. We clarified the critical elements to implement the multi-domain translation system and verified reliability and effectiveness by comparing the results of proposed approaches and baseline. We will also look into new research directions for multi-domain translation in the future.

Chapter 2

Proposed Framework

In this chapter, we present the framework of a multi-domain translation system that relies on a model selection module and a model ensemble module. The former module is devised to adequately choose relevant domain-specific models, given a source sentence. The latter module, on the other hand, is designed to effectively combine chosen models to decode (generate) each word in the to-be-generated target sentence. Figure 2.1 illustrates the proposed framework.

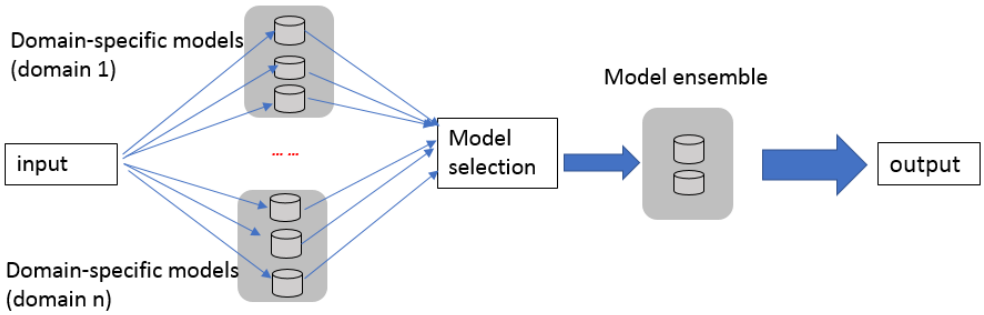


Figure 2.1: Proposed framework of multi-domain translation system

When an input sentence is entered into the system, it is forced to select relevant domain-specific models from the array of already-trained domain-specific models. Note here that each domain can have multiple domain-specific models, and each of them is trained under different conditions. Each individual domain-specific translation model should guarantee a considerable degree of translation quality for

the designated domain.

Four specific objectives (or steps) have to be implemented to realize the proposed framework of a multi-domain translation system: They are (1) Preparation of corpus resource, (2) Model training, (3) Model Tuning, and (4) Model selection and ensemble.

- Preparation of corpus resource

Corpora are widely used in dictionary compilation, language teaching, traditional language research, and natural language processing research. In the specific research field of data-driven machine translation, a corpus is the primary requisite for training and evaluating translation models.

The text in the corpus is usually language material that appeared through natural use. In order to facilitate reliable processing, text in the corpus is usually processed according to some actual research needs. For example, for Japanese syntax analysis, text in the corpus should be segmented into words.

In this thesis, we describe how to prepare a corpus resource for both training the translation model and evaluating the translation system. As a preliminary preparation, we need to consider what resources will be used to train the model. When collecting these resources, the following points should be considered.

Domain: We need to prepare a set of domain-specific corpora to build a multi-domain translation system so that relevant domain-specific translation models can be chosen and ensembled.

Language: According to the language(s) of a corpus, the type of corpus can be classified either of monolingual (one language), bilingual (two languages), and multilingual (multiple languages). From the perspective of training a data-driven translation model, we need to have a bilingual parallel corpus.

Alignment unit: According to the aligned unit of the corpus, the type of corpus can be divided into text-based, sentence-based, and phrase-based. In this

thesis, we only focus on corpora aligned at the sentence level, since the training of an NMT model requires a corpus to be aligned at the sentence level.

Text quality: The use of a corpus with low text quality affects the quality of the trained translation model. It is thus required to prepare a corpus accommodating high-quality texts.

Corpus size: The training of an NMT model requires large-sized corpora. That is, a translation model trained with a larger corpus generally performs better than that trained with a smaller corpus. In order to acquire as much bilingual corpora as possible，in addition to collecting existing corpora, we also need to consider utilizing monolingual corpora to expand the existing bilingual ones.

- Model training

To realize the proposed framework summarized in Figure 2.1, we need to train multiple domain-specific translation models. In order to train a domain-specific translation model, we adopt an approach based on domain adaptation, which will be detailed in Chapter 3. For each and every translation domain, the corpus utilized during the training can be divided into two types: an out-of-domain corpus that collects texts from domains other than the present domain, and an in-domain corpus which is composed of the texts collected from the present domain. By employing the methods of domain adaptation, we can achieve translations more tailored to the present domain by using an in-domain corpus, while maintaining the generality of translations by referring to out-of-domain corpora.

Domain adaptation is accomplished by adopting some existing approaches, such as Fine-tuning, Mixed Fine-tuning, stacking, as well as a novel approach referred to as data selection. By using these approaches for domain adaptation, it is expected to achieve a good translation model tailored to a specific domain. Among these approaches, the newly invented data selection method first runs an alignment

algorithm to select semantically similar sentences from an out-of-domain corpus, and then adds these sentences to the in-domain corpus.

For each and every domain considered, we train a number of domain-specific models. To be more specific, each of the models is trained by adopting one of the domain adaptation approaches. Thus, the number of models for a domain amounts to the number of domain adaptation approaches.

The details of these approaches to domain adaptation will be discussed in Chapter 3.

- Model tuning

Once an initial translation model is built, we then need to evaluate the model to improve it. The process of model improvement is usually referred to as model tuning.

Generally, the low translation qualities are primarily attributed to the issues found in the training corpora, including the low quality or low diversity of the training corpus, and the imbalance between the in-domain corpus and the out-of-domain corpus. Besides, general issues in machine translation also affect the translation qualities. These include:

- Unknown words: Some words in the input are not contained in the system's vocabulary, causing the words to be left untranslated in the output.
- Erroneous translations: Some words in the input are translated into sometimes totally irrelevant words in the output. This issue is prominent in NMT, rarely observed in SMT.
- Inconsistent translations: A word occurring at multiple places but with the same meaning in the input text is translated inconsistently in the output text. This issue is also typical in NMT, where the translation of a word in the input is selected on purely statistical grounds depending on the surrounding words.

The issues in model tuning and the solutions to address them will be discussed in Chapter 5.

- Model selection and ensemble

The proposed framework for realizing a multi-domain translation system, as depicted in Figure 2.1, is most characterized by its *select-and-ensemble* approach. With this approach, given an input sentence, some relevant domain-specific models are first selected, and then the words in the generated output are decoded by choosing the most plausible translation candidate yielded by the chosen models. By doing so, our system can translate input sentences that are somehow domain-specific as much as tailored to them, while maintaining generality towards domain-neutral sentences.

We use a perplexity-based evaluation metric to select relevant domain-specific translation models that best match the content of an input sentence. As described in Section 4.2, perplexity is a frequently used metric to assess the quality of a language model. We employ this metric in the proposed method as a measure to estimate the difficulty of translation with a specific translation model. The proposed model ensemble method refers to the outputs of the selected domain-specific models and chooses the most plausible word. We devised several ways to calculate the likelihood of a translation candidate, and experimentally compared them.

The entire translation process of the proposed framework does not require the domain of an input to be explicitly predicted thanks to the *select-and-ensemble* approach. The results presented in Chapter 7 empirically show the efficacy of our *selection-then-ensemble* approach.

Chapter 3

Domain Adaptation of the Models

This chapter discusses approaches to adapting a translation model to a specific domain.

In Chapter 2, we mentioned that the proposed framework requires a certain number of domain-specific translation models. Each of these domain-specific translation models is specialized for a specific domain to achieve more suitable translations in that domain compared to those attained by a general-purpose translation model.

Training a sufficiently good domain-specific translation model requires a large number of texts in the in-domain corpus. However, the size of an available domain-specific corpus is generally small in actual cases. A plausible approach to deal with this situation is the adoption of domain adaptation techniques.

Domain adaptation is a research area in machine learning. In particular, it is deeply associated with transfer learning. The central idea of domain adaptation is to tailor a model already trained in other domains to the desired target domain. Once the model is successfully transferred to the target domain, it can deal with domain-specific data while maintaining generality towards domain-neutral data. Thus, in the field of MT, domain adaptation can overcome the limitation of an insufficient amount of texts in an in-domain corpus. We review related work on domain adaptation in the field of MT in Section 3.2.

In Section 3.3, we specifically introduce five domain adaptation methods to

train a domain-specific translation model. We can prepare five different domain-specific translation models for each domain of interest. These domain-specific models are used as candidate models for model selection, which will be described in Chapter 4.

3.1 Domain Adaptation

A general-purpose translation system often performs poorly in a specific domain, as reported in (Koehn & Knowles, 2017), which highlights the importance of developing a domain-specific translation system. High-quality domain-specific machine translation systems are in high demand, as the performance of a general-purpose MT system is generally limited.

However, as the amount of available domain-specific texts in a specific domain is usually limited, it is difficult to train a domain-specific translation system with a plentiful amount of in-domain text.

The literature, such as (Duh et al., 2013; Koehn & Knowles, 2017; Sennrich et al., 2013; Zoph et al., 2016), pointed out that the models trained with low resource domain-specific data perform poorly in translating domain-specific texts.

The approach of domain adaptation comes into play to remedy the issue of insufficient domain-specific data (Chu et al., 2018; Wang et al., 2016). Note that the corpus prepared for a specific domain is called an in-domain corpus, whereas the corpus collecting texts collected from other domains is referred to as an out-of-domain corpus. In many cases, the size of texts in an in-domain corpus is generally small compared to that of an out-of-domain corpus.

The approaches of domain adaptation have been effectively applied to both SMT and NMT. The following Section 3.2 will detail the existing domain adaptation approaches in NMT, which are roughly grouped into data centric approaches and model centric approaches.

3.2 Related Work in MT

The data centric approach mainly focuses on the selection of sentences from the out-of-domain corpus to expand the in-domain corpus by using a variety of methods. A typical method is to use trained language models to measure the similarity between an out-of-domain sentence and an in-domain sentence, and then to choose those scoring over a predefined threshold to expand the in-domain corpus. This method is useful, as text irrelevant to the target domain would decrease the translation quality in the domain. From the viewpoint of data augmentation[1], it is crucial to devise an excellent method to select proper sentences from an out-of-domain corpus.

Model centric also focuses on modifying the model training process. With this approach, however, the out-of-domain corpus and the in-domain corpus are separately utilized as a training corpus at different steps during the entire training process.

Most of the domain adaptation approaches are inspired by the existing methods developed in SMT research.

3.2.1 Data Centric

Zhang & Zong (2016b) use source-side monolingual data from out-of-domain data to enhance the NMT encoder through multi-task learning to obtain two models. The first model is a translation model built for predicting translated target-side sentences, and the other one is devised for reordering the source-side sentences. Their expectation for this multi-task learning regime is that by jointly predicting the optimal reordering of the source sentence one could enhance training to achieve a better translation model.

Cheng et al. (2016) reconstruct out-of-domain monolingual data on both sides of the source and target in order to train a better translation models. For recon-

[1]Data augmentation is a popular topic in machine learning, as there is a concrete principle if applied under the right circumstances: More data is always better.

structing out-of-domain data, they utilized the NMT as an autoencoder. [2]

Sennrich et al. (2015) reasonably expected that because NMT can learn a language model, the target out-of-domain monolingual data can also be used in NMT systems. More specifically, they proved that the synthetic parallel corpus generated through the reverse translation of the target sentences could contribute to enhancing the decoder. In contrast to Sennrich et al. (2015), Zhang & Zong (2016a) and Park et al. (2017) have shown that the target-side monolingual data, the source-side monolingual data, as well as the generated synthetic data are effective in domain adaptation.

Wang et al. (2017a) proposed to use sentence vectors to select adequate sentences from the out-of-domain corpus. Their method measures the similarity between an in-domain sentence and an out-of-domain sentence by using their sentence embedding vectors. The most similar out-of-domain sentences are then selected to expand the in-domain corpus. Here a sentence embedding is computed as a centroid of word vectors appearing in the sentence.

Van der Wees et al. (2017) proposed a dynamic data selection method. It split the training process into different stages. During the training process, different in-domain data would be selected to expand the out-of-domain data at different training stages.

Chu et al. (2017) and Chu et al. (2018) improved the method originally proposed by Sennrich et al. (2016a). The improved method uses specially-designed tags to control the training of NMT. In this method, corpora of multiple domains are combined together with two small modifications: using tags and oversampling the corpus. Using tags will append the domain tag "<2domain>" to the source sentence of the corresponding corpus. The domain tag would help the decoder of the NMT system to generate sentences for a specific domain. Another modification, oversampling the corpus, will enlarge the size of the in-domain corpus to be as large as the out-of-domain corpus. It causes the training process to pay almost

[2] An autoencoder is a special type of neural network which is trained to reconstruct the input data as output. By doing so, it could be expected that the hidden layers of a network could learn to represent the essential features of inputs in the lower-dimensional vectors.

equal attention to the in-domain corpus and the out-of-domain corpus.

Sajjad et al. (2017) examined the preferable order of training by comparing different methods in training multi-domain translation systems. Their results demonstrated that a translation model should be first trained by a large out-of-domain corpus and then fine-tuned by a smaller in-domain corpus.

3.2.2 Model Centric

The prominent difference between model centric and data centric is that model centric modifies the training process to obtain the best translation model for in-domain data.

Luong & Manning (2015) proposed a two-step training method for domain adaptation. In the first step, the translation model is trained with the out-of-domain data. Then in the second step, training of the model obtained from the first step continues with the in-domain data. After its introduction, Sennrich et al. (2015), Servan et al. (2016), and Freitag & Al-Onaizan (2016) have all adopted this idea. Their methods first trained an NMT model with a rich out-of-domain corpus until it converged, and then started a second training process using the first model as the initial model. In the second training process, it is expected that the initial model could provide proper initial model parameters to start with, and hence ensuring better model tuning. Generally, this kind of training regime is called fine-tuning and is recognized as a common technique in domain adaptation.

Fine-tuning of a model could cause degradation of the quality of the translation of out-of-domain or domain-neutral sentences. In order to prevent the degradation, Dakwale (2017) proposed an extension of fine-tuning, which can maintain the distribution of out-of-domain data during the fine-tuning based on the utilization of knowledge distillation[3] technique.

Miceli Barone et al. (2017) attempted to solve the problem of overfitting[4]

[3]Knowledge distillation refers to a technique that uses the knowledge obtained from a bigger model to improve the performance of a smaller model.

[4]Overfitting is a critical problem in machine learning. It represents a situation where the trained model is too close to the training data, and hence loses generality to predict the test data.

which has to be avoided during fine-tuning. Their strategy to solve the overfitting problem is to explore regularization[5] techniques, such as dropout[6] and L2 regularization[7]. In addition, they considered the method of tuneout, which is a variant of standard dropout[8].

On the other hand, Wang et al. (2017b) proposed a method for adjusting instance[9] weights, which increases the domain instance loss proportion when calculating the entire loss function[10]. Wang also proposed a method for adjusting weight parameters. During the training process, the out-of-domain and in-domain data are jointly trained, and the ratio of out-of-domain sentences to in-domain sentences in each batch of mini-batch[11] training is adjusted. As a complement, Wang et al. (2018) proposed a joint framework for sentence selection. It measures the degree of correspondence between a sentence and an available entire domain dataset by introducing a domain similarity metric.

Kobus et al. (2016) proposed to add the domain tag as an additional feature into the word embedding vector, which could exploit the domain information

[5]In essence, regularization introduces a kind of penalty to the loss function used in training. By doing so, it is possible to prevent the situation where model parameters become too large, which leads to overfitting. It contributes to increasing the generalization ability of the model.

[6]Dropout: Randomly ignoring some nodes in a neural network during each update step of the training process to prevent nodes from co-adapting too much and thereby improving the generalization performance of the model. More details in Srivastava et al. (2014).

[7]L2 regularization: One type of regularization for preventing the number of parameters from becoming too large. Also known as Ridge regularization.

[8]Standard dropout method sets the weight of randomly selected nodes to zero to enhance the robustness of a model, while tuneout sets them depending on the corresponding columns of related parameters given by out-of-domain data. More details are provided in (Miceli Barone et al., 2017).

[9]Instance refers to the basic unit of the training data in machine learning. For example, when training an NMT model, we need to input sentence pairs. In this case, one sentence pair is one instance.

[10]The loss function, or objective function, is essential in training. It measures the difference between the parameters based on the current model and the parameters based on the ground-truth data. The process of training can be seen as a parameter optimization process that tries to minimize the differences computed by the loss function.

[11]Mini batch: Updating the model parameters uses only a portion of the data on each update. That portion of data is called a batch. A mini-batch refers to a set of a small number of batches. It is a common strategy to use mini-batch training rather than instance-based training or whole-batch training. The mini-batch training contributes to improving the training time while consuming less memory.

at the word level. They also proposed a method based on the term frequency-inverse document frequency (tf-idf) measure to predict the domain tag of the input sentences.

Apart from the method of fine-tuning, Cho et al. (2014) also trained an RNN language model on monolingual data. Then they combined the RNN language model with NMT models. In combining these models, the next word generated by the NMT model is assumed to be restated by a weighted sum of the probabilities yielded by the NMT model and RNN language model.

Freitag & Al-Onaizan (2016) proposed to ensemble out-of-domain models with the fine tuned in-domain models. Khayrallah et al. (2017) proposed a stack-based word lattice decoding algorithm. In domain adaptation experiments, the results show that stack-based decoding can achieve good results.

Britz et al. (2017) proposed a domain discriminator to leverage the diversity of information in the multi-domain corpus. In their method, they added a feed-forward network (FFNN) as a discriminator onto the model framework. The results show that their method can improve the accuracy of domain prediction. These good results can be attributed to the encoder using an attention mechanism to predict the domain of the source sentence so that the discriminator can be optimized together with the translation model.

3.3 Train Domain-specific Model Using Domain Adaptation

We describe the domain adaptation methods used in this thesis. Five different domain adaptation methods were used to train the domain-specific model. They are referred to as ALL, fine-tuning, mixed fine-tuning, stacking, and data selection.

3.3.1 ALL

A basic method for training a neural domain-specific translation model from Sennrich et al. (2016b) is to use all available corpora in each domain simply. At

first, we need to combine all bilingual parallel data before training the system. During training, an in-domain validation set is used to guide the training of the right model parameters. The resulting model has the advantage of seeing a mix of all available data so that it can stably handle heterogeneous test data.

In this thesis, we not only combine in-domain and out-of-domain corpora, but also try to adjust the ratio of texts from each domain after combination.

For the experiment, which will be discussed in Chapter 7, in-domain data is usually on the level of millions of sentences, while out-of-domain data is in the tens of millions. The details of the corpus data will be shown in Chapter 6. This difference in the ratio will significantly affect the translation performance of the model trained by the ALL method.

The specific adjustment is to oversample the in-domain data from a small size to a prescribed large size. The golden ratio of oversampling will vary depending on the specific domains. By adjusting the ratio of the in-domain corpus to out-of-domain corpus, we can get a BLEU score[12] increase of more than 10 points. Regarding the impact of the proportional adjustment on the ALL model, a more detailed description and specific analysis will be given in Chapter 5.

3.3.2 Fine-Tuning

As already discussed, since Luong & Manning (2015) trained an initial model on out-of-domain data and later continued to train the initial model on in-domain data, this idea has become a common method of performing domain adaptation. In this way the parameters of the final model are tuned towards the in-domain data. This approach is referred to as fine-tuning， from which many variants with various setting standards have been developed.

In this thesis, we use the most basic setting standards inspired from Servan et al. (2016) as a representative of the fine-tuning method to train the domain-specific

[12]BLEU is a metric to measure the quality of translation. It computes the degree of n-gram overlaps between a machine-generated translation and the corresponding reference translations. The detail of BLEU will be introduced in Section 7.2.1.

model.

In fine-tuning, the main difference in configuration between a training model (out-of-domain) and another training model (in-domain) is the utilization of a training corpus, as shown in Figure 3.1. We need to consider both in-domain data and a large amount of out-of-domain data during the training process. The training process in the first step only considers out-of-domain data, and then during the retraining phase, only a small amount of data in the in-domain domain is used for the specialization process. The main purpose of this training configuration is to simulate an incremental adaptation framework that enables the adaptation process only when in-domain data is available.

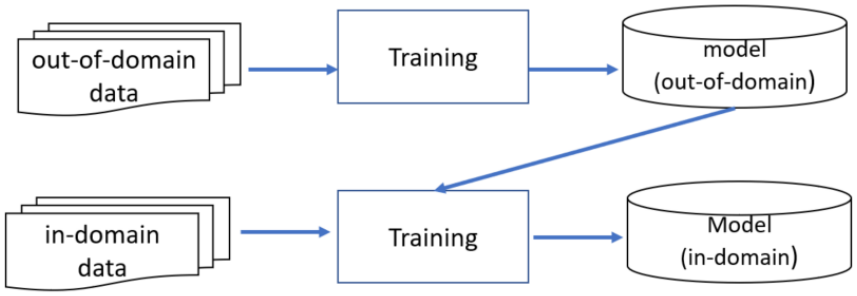


Figure 3.1: Fine-tuning

In the following subsections, we also introduce a method of mixed fine-tuning and of stacking, the two variants of fine-tuning used to train the domain-specific model.

3.3.3 Mixed Fine-Tuning

For the third method, we adopted the method of mixed fine-tuning proposed by Chu et al. (2017). This method is the first variant of the fine-tuning method.

Different from the training process of fine-tuning, the mixed fine-tuning is more elaborate, as shown in Figure 3.2.

There are four main steps.

- Step1 is training an NMT model (out-of-domain) using out-of-domain data.
- Step2 is merging in-domain and out-of-domain data (by oversampling the in-domain data).
- Step3 is training the NMT model (mixed) from step 1 using the merged data.
- Step4 is training the NMT model (in-domain) from step 3 using in-domain data.

One purpose of using mixed fine-tuning is to solve the overfitting problem, which is caused by the in-domain data being too small.

For mixed fine-tuning, after training out-of-domain data obtain a good model, the training process is tuned using a mix of in-domain and out-of-domain data, which can prevent overfitting brought about when only using in-domain data.

The research of Chu et al. (2017) shows that the effect of mixed fine-tuning is better than plain fine-tuning. Because the mixed training corpus in the second training step (Step2) includes the training corpus (out-of-domain data) in the first training step (Step1), this makes it faster to converge when training such mixed domain training data.

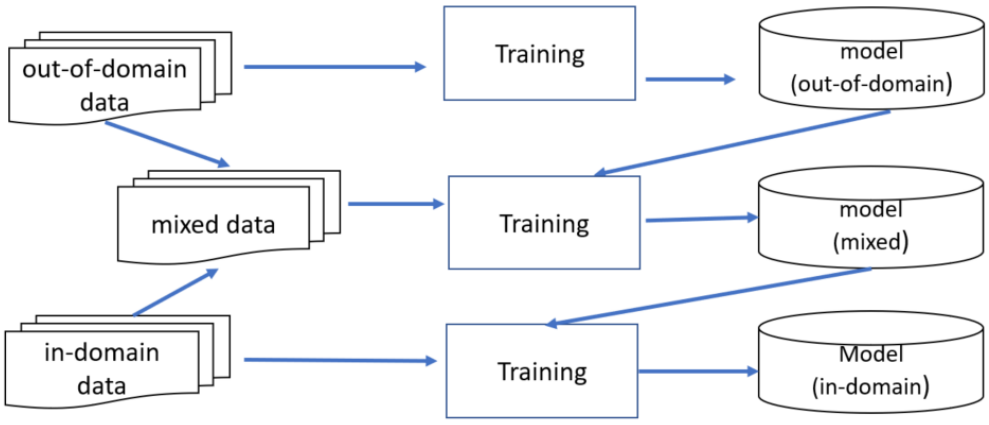


Figure 3.2: Mixed fine-tuning

3.3.4 Stacking

As the fourth method of domain adaptation, we refer to and use the method of stacking, another type of fine-tuning variant proposed by Sajjad et al. (2017).

Sajjad et al. (2017) adopt a multi-step execution method with fine-tuning at each step to train the system. As Figure 3.3 shows, data (out-domain1) and data (out-domain2) represent the first and second subset of out-of-domain data, respectively. data (out-domain1) would be used for training in the first to N-th epoch[13]. Until the M-th epoch, data (out-domain2) is used for training. In the last few epochs, D_i is used for training. In this method of domain adaptation, the model will learn from multiple data sets one by one, so this method is named stacking.

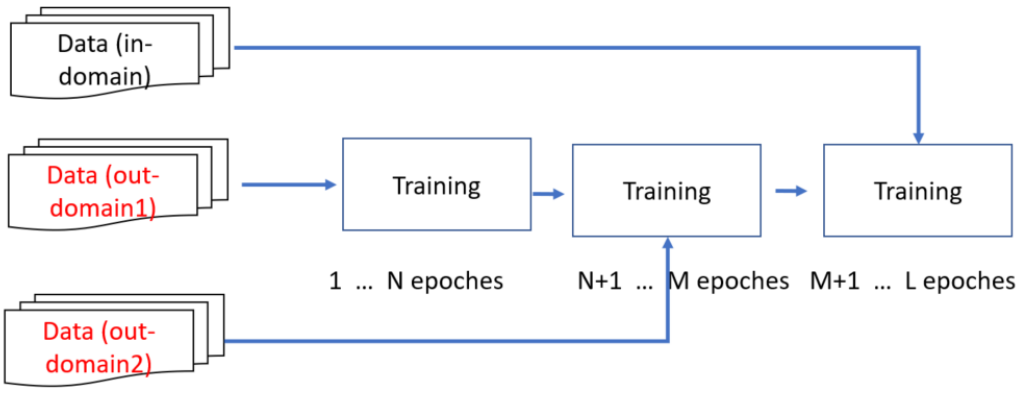


Figure 3.3: Stacking

Considering that the out-of-domain corpus is much larger than the in-domain corpus, we have to pay attention to the problem of imbalances in quantity between the out-of-domain corpus and the in-domain corpus. For example, in this thesis, the size of the out-of-domain corpus can reach more than ten times the in-domain corpus. The stacking method can avoid the problem of size imbalances between

[13]Epoch: Each time model parameters are updated during the training process, only a part of the training data may be used in turn. An epoch refers to a training process in which all parts of the training data have been used.

out-of-domain and in-domain corpora because, for a particular training instance, out-of-domain and in-domain data will not be used at the same time. It is also an important feature that distinguishes this method from a traditional fine-tuning method.

Regardless of the method adopted, we notice that adding different domain data to different stages of the training process can help the model perform more flexibly on in-domain data. In addition, it should be noted that the report of Sajjad et al. (2017) also mentioned that the order of each domain in the entire stacking process has a great impact on the final translation model.

As shown in the results of Sajjad et al. (2017) reproduced in Figure 3.4, when the UN data is trained before the Opus data, the performance (UN2Opus in Figure 3.4) is significantly improved. Conversely, if training Opus data at first and then training UN data, the performance of the model (Opus2UN in Figure 3.4) will decrease. In response to this phenomenon, we adopted a shuffling process to avoid using the training corpora in a fixed order. We shuffle all out-of-domain data, and then divide it into subsets of out-of-domain data.

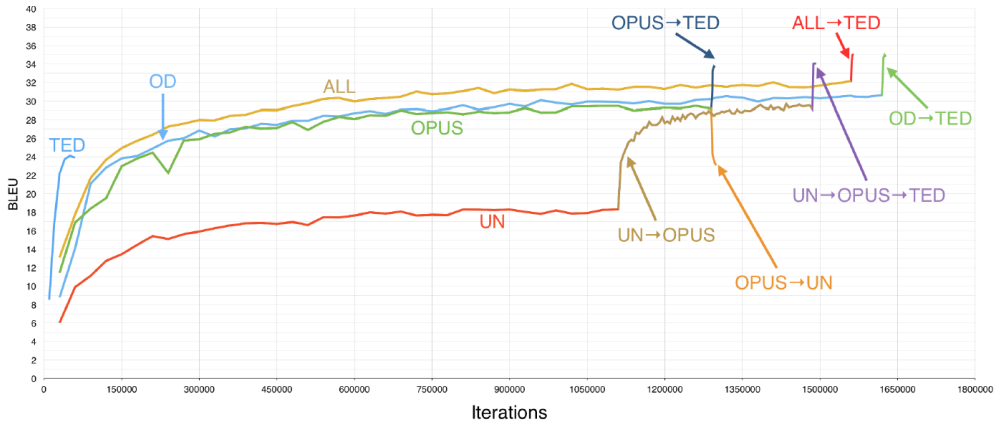


Figure 3.4: Training order: Opus2UN vs. UN2Opus
(adopted from Sajjad et al. (2017), figure 2)

3.3.5 Data Selection

The method of fine-tuning and its variants belong to the idea of the model centric approach, but we also propose a method of data selection based on the idea of the data centric approach.

Using all the available data to train the model can be computationally expensive, such as for the ALL method mentioned earlier. In particular, a translation-based method is used to generate bilingual data from monolingual data, thereby expanding the in-domain data to the order of 100 million sentences, which takes more than two days for each training on a computer with 4 GPUs. This amount of computational overhead cannot be ignored. In addition, when the in-domain data needs to grow to the same amount as the out-of-domain data through oversampling, the cost of a single training is close to twice that of other methods.

Here an alternative method could be proposed, that is, data selection. There are two advantages to data selection. The first one is to reduce training costs. The second one is to avoid using data that is too dissimilar to the in-domain data, since it may be less effective for training domain-specific models.

In this thesis, we will consider selecting a part of out-of-domain data, merging this part of data with in-domain data, and then training a model with the combination. With good criteria, choosing some appropriate out-of-domain data for training leads to a better domain-specific model.

For the criteria of data selection, researchers usually tend first to create a language model and then use this language model to obtain some statistical indicators as a reference for selecting the appropriate out-of-domain data. For example, Axelrod et al. (2011) use an n-gram model and cross-entropy. In addition, Durrani et al. (2015) explores data selection using joint neural networks. Liu et al. (2018) proposed a reinforcement learning (RL) framework that can dynamically search out-of-domain data related to in-domain data. Their experiments prove that this framework can be effectively applied to different NLP tasks.

In contrast to the above-mentioned criteria, in this thesis, we refer to a criterion

based on semantic similarity from Luo & Hayashi (2015), and improve the method (as seen in Figure 3.5) for the specific implementation of this thesis. Luo & Hayashi (2015) used word-level alignment information to calculate an alignment score. So the alignment score can be calculated and used to measure the semantic similarity between sentences. The calculation formula of the alignment score is presented as Equation 3.1.

$$score(S_1, S_2) = \frac{2 \times prop_1 \times prop_2}{prop_1 + prop_2} \tag{3.1}$$

The two values $prop_1$ and $prop_2$ represent the proportion of words in sentence 1 and sentence 2 that have a word-level alignment or correspondence. These values, $prop_1$ and $prop_2$, are calculated by dividing the number of words with word-level alignment by the total number of words in the sentence itself.

In the scenario of this thesis, for a bilingual pair, the alignment score can be calculated between two Japanese sentences, or between English or Chinese sentences.

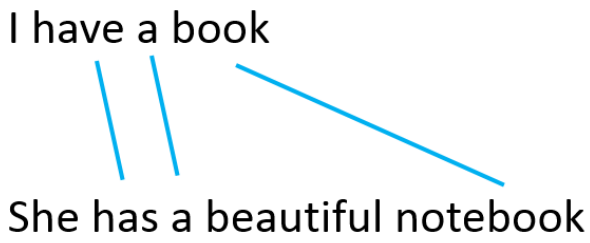


Figure 3.5: Alignment example

Here we can give a specific example to calculate the alignment score. Assume that between two English sentences, some words have a corresponding relationship. (Like the words connected by the blue lines in Figure 3.5.)

Then, correspondingly, the number of words with word-level correspondences in sentence 1 is 3: “have” “a” “book”. As there are four words in sentence 1, $prop_1$ is 0.75.

On the other hand, the number of words with word-level correspondences in sentence 2 is 3: “ has ” “ a ” “ notebook ”. There are five words in sentence 2, so $prop_2$ is 0.6.

According to Equation 3.1, we know that the final alignment score will range from 0 to 1. In this example, we calculate that the value of the alignment score is 0.66.

While selecting from out-of-domain data, these corpora must be suitable for use in the in-domain setting. Therefore, we use the concept of semantic similarity to select corpora. Previously, Luo & Hayashi (2015) used this equation 3.1 as a measure of semantic similarity between sentences.

However, since the primary purpose of this thesis is not to obtain an accurate semantic similarity, which can find data most similar to that in in-domain corpora, we made a few changes to this measure. As our objective is to select a sufficient number of data from out-of-domain corpora, if the semantic similarity is too accurate, too few corpora will be selected. So we have improved the above method in Luo & Hayashi (2015).

Because we need to perform a large number of comparisons, if each sentence is aligned, this will incur a relatively large computational overhead. For judging whether two words are aligned, we directly use the word embedding vector based on Wikipedia[14] to calculate the similarity between words. If the similarity of a pair of words exceeds a certain threshold, then we assume that there is a corresponding relationship between the pair of words. In this way we can get an alignment score without performing an actual alignment.

In addition, in order to adjust the number of available out-of-domain corpora, we can increase or reduce the size of the data selection results by adjusting the threshold for determining similarity. The threshold depends on the size of the

[14]We generate word embeddings from Wikipedia dumps in each of the languages used: Japanese, English, and Chinese.
Wikipedia (en): https://dumps.wikimedia.org/enwiki/
Wikipedia (zh): https://dumps.wikimedia.org/zhwiki/
Wikipedia (ja): https://dumps.wikimedia.org/jawiki/

corpus and the ratio of the in-domain corpus to out-domain corpus. The specific description will be given in Section 5.2.2.

3.4 Chapter Summary

Domain adaption is an effective approach for training domain-specific models, which make full use of the in-domain and out-of-domain corpus. By using the five domain adaptation methods introduced in Section 3.3, we can train five types of domain-specific models. In Chapter 5, we will additionally examine some of the actual translation problems that occur in these domain-specific models, and tuning models to achieve a better translation performance.

Chapter 4

Multi-domain Translation

This chapter will discuss how to use the domain-specific models prepared in Chapter 3 for multi-domain translation.

Section 4.1 will briefly describe the translation performance of these domain-specific models trained by the five domain adaptation methods proposed in Chapter 3. Following the creation of the candidate models, Section 4.2 will discuss the approach taken to perform the model selection. The purpose of model selection is to determine which domain-specific translation models are most suitable for translating the input text, even without utilizing domain information. In order to achieve this purpose, we choose a measure based on perplexity to estimate the translation difficulty of translating the input text for each candidate model. The model with less measured translation difficulty is considered as the better model for translating input text. Finally, Section 4.3 will discuss the model ensemble approach. After selecting the appropriate models for input text, we combine the selected models to output translation results by weighting these models with two different methods.

In short, the process of multi-domain translation can be simply concluded as two main steps, model selection for choosing the appropriate model from candidate models as described in Section 4.2, and model ensemble for synthesizing each selected model as described in Section 4.3.

4.1 Candidate Models

We apply the approaches of model selection and model ensemble to our proposed translation system. The idea of model ensemble is adopted from machine learning, that is not just to build one model for a problem, but instead build multiple ones and then combine them. That is why it is referred to as model ensemble.

Candidate models are the premise of a successful ensemble. In order to prepare enough candidate models for an ensemble, a simple strategy is to train candidate models using different subsets of the training data. For neural networks, an alternative strategy is to collect candidate models with different initializations or models trained with different epoch counts.

In Chapter 3, we discussed domain adaptation methods for training domain-specific models. These models will be considered as the candidate models in this chapter. Because we use five domain adaptation methods, we can build five types of candidate domain-specific models for each specific domain.

Each model may have its strengths or weaknesses. With specific input, it is difficult for us to guarantee that the translation given by a single model is correct. Although each model may make different mistakes, we can refer to multiple models to avoid such mistakes. Unless all candidate models are ineffective, it is better to utilize multiple models instead of one model when generating translations.

Before moving on to the next section about describing model selection, we show the quality of translation using these candidate models independently.

Tables 4.1 and 4.2 record the translation quality of the language directions Japanese-to-English (ja-en) and Japanese-to-Chinese (ja-zh), respectively. The columns in the two tables, ALL, fine-tuning, mixed fine-tuning, stacking, and data selection, correspond to the domain adaptation method used. As a comparison, there is an additional column named baseline model. This model is obtained simply by training without any domain adaptation methods. Hence the baseline model has not been tuned for any domain. The rows in the two tables, News, Travel, and IT, as their names suggest, each represents a test set for the specific

Table 4.1: Domain-specific models (ja-en)

	baseline	ALL	fine-tuning	mixed fine-tuning	stacking	data selection
NEWS	37.1	41.57	39.34	40.09	37.87	39.24
Travel	26.63	29.26	31.01	32.67	29.30	30.76
IT	22.46	33.89	29.94	35.07	31.68	35.39

Table 4.2: Domain-specific models (ja-en)

	baseline	ALL	fine-tuning	mixed fine-tuning	stacking	data selection
NEWS	36.83	39.45	39.14	39.24	38.87	37.90
Travel	25.46	26.93	27.22	28.08	26.48	27.06
IT	20.08	28.38	32.02	34.67	31.87	32.16

corresponding domain. The values in cells are the BLEU scores，which are widely used in machine translation evaluation. For example, the BLEU of the model using the ALL adaptation method on the News test set in Table 4.2 is 39.45, which is the best one among the result row of the News test set. By looking at the two tables, we can roughly observe the trend of translation performances when these candidate models operate independently without model selection and model ensemble.

Since higher BLEU scores mean better translation results, it can be considered that on the News test set, the model using the ALL adaptation method is more effective than the model using the stacking adaptation method (39.45 is higher than 38.87).

A more detailed description of the test dataset and machine translation evaluation BLEU will be presented in Chapter 7 later. The purpose of showing tables with numerical values here is to explain how effective these candidate models are briefly.

First of all, compared with the results of the baseline, we can see that by applying the method of domain adaptation, each candidate model is indeed improved on in the specific domain. The efficacy of these candidate models is confirmed.

However, at the same time, through a horizontal comparison of the models of five types of domain adaptation methods, we quickly discovered that it is difficult to find out which candidate model is the best. As a result, we cannot determine

the most effective domain adaptation method for translating multi-domain text. For example, referring to Table 4.1, if the domain of the test set is News, the best model uses the mixed fine-tuning method. Nevertheless, if the domain of the test set is IT, the best model is trained with the data selection method.

Hence, the significance of using multiple candidate models for multi-domain translation is highlighted by these results of Tables 4.1 and 4.2. The two main parts of multi-domain translation, model selection and model ensemble, will be introduced in following sections. The premise of the successful execution of these two parts is that we have provided effective candidate models.

4.2 Model Selection

Since the candidate models developed in Chapter 3 are confirmed to be effective, in this section, we discuss how to select the most suitable candidate model for a particular input text.

We adopt perplexity as a measure to evaluate and select a candidate model. For this thesis, the perplexity measure represents the difficulty of the candidate model in translating an input sentence. A model which has the least difficulties in translating an input sentence should be considered effective. Therefore, the candidate model with the lower perplexity measure would be selected.

Perplexity is widely used as an essential metric in other NLP research areas. This related research has shown the significance of using perplexity metrics. For example, perplexity is often used as a criterion for judging the quality of a language model, such as the work of (Chen & Goodman, 1996).

Furthermore, in order to train recurrent neural network (RNN) language models, Fernandez & Downey (2018) proposed a sampling approach based on an unsupervised approach. This sampling approach will preferentially select sentences with higher perplexity to join the training set. A better model could thus be developed by training with more difficult sentences.

When Brown et al. (1993) and Al-onaizan et al. (1999) were studying SMT,

perplexity was used to evaluate the models with different model training settings. The results showed that as the size of the training corpus increased, the model perplexity of translating the test set decreased. In other words, the larger the training corpus is, the better the translation model is.

Gamallo et al. (2017) described a task of Similar Languages Discrimination[1]. In their study, perplexity is utilized as a measure of how well a model fits the test data. The basic strategy of the system employed to accomplish this task is using the language distance calculated by the mean number of model perplexities.

For Hybrid Multi-System Machine Translation, Rikters (2016) used perplexity to measure the sentence chunks translated by multiple systems. More specifically, they split the input sentence into chunks, selected translated chunks from the results of multiple systems based on perplexity measure, and combined the selected chunks into a final translation result.

As early as the research on domain adaptation of SMT, Sennrich (2012) proposed a method of weighted combination, which combined the corpus depending on weighted counts. In order to measure the best weight, they proposed perplexity optimization for weighted counts, where the best weighted counts should minimize the perplexity measure.

For model selection, there are two obvious reasons to utilize perplexity. The first reason for using the perplexity measure is that it can help us to estimate the difficulty of a candidate model in translating an input sentence. It is a simple and effective measure for selecting models. Secondly, it requires simple calculations. Unlike other measures, such as BLEU, which require an additional comparison between the translation result and reference sentence after the translation process, the calculation of perplexity itself can be included in the translation process of using the candidate model.

Perplexity is calculated as Equation 4.1:

[1]Similar Languages Discrimination focuses on discriminating text between similar languages. For example, the similar languages could be Bosnian-Croatian-Serbian or Malay-Indonesian.

$$Perplexity = b^{-\frac{1}{N}\sum_{N}^{i=1} log_b\, q(x_i)} \tag{4.1}$$

Here, N represents the total number of words in the input sentence, and $q(x_i)$ represents the translation probability of the i-th word (unigram). Here, the value of b is set to e (base of the natural logarithm).

From the equation, we can recognize that if the sentence translation probability is not high enough, then the value of perplexity will be relatively large. This shows how difficult it is for the NMT model to translate the input sentence. Thus we should try to choose models that can easily translate the input sentence. We sort all candidate models according to the perplexity measure and select models with smaller perplexity values. In this way, we have selected the most appropriate models for the input text.

One advantage of using perplexity as a reference measure is that it can be obtained during the translation process without making a comparison with the corresponding reference sentence. For example, if we use BLEU as the measure of translation quality to select models, there must be a reference sentence to compare with the translation. But, in fact, we may not have a corresponding reference sentence for the input text.

4.3 Model Ensemble

After we use the perplexity-based method to select the appropriate model for input sentences from candidate models, we must synthesize models to generate the output in the final step of the translation process. This process is called the "model ensemble".

Model ensemble is a common technique in machine learning. Its purpose is to not only build a system, but also to build multiple systems and then combine them together. Each model behaves differently when faced with different texts. But when most models make the same translation for a text, the credibility of the translation will be high. This voting-like principle is widely used in machine

learning.

In this section, we first review the related research concerning the use of model ensembles in NMT and SMT, and then introduce the model ensemble method used in this thesis.

4.3.1 Related Work for Model Ensemble

Researchers first experimented with many model ensemble methods on SMT. Some studies, like (Och & Ney, 2001), (Matusov et al., 2006), (Schwartz, 2008), and (Schroeder et al., 2009) have successfully applied the model ensemble to SMT tasks. However, SMT generally has complex structures, such as PBSMT (Phrase-Based SMT, Koehn et al. (2003)) and HSMT (Hierarchical phrase-based SMT, Chiang (2007)). Therefore, the ensemble applied to SMT is usually limited to a combination of n results when choosing the best result. Rokach (2009) further proved that combining multiple systems can significantly improve performance.

When moving the focus to NMT, there are also many model ensemble methods. Most current NMT frameworks, such as those proposed by Sutskever et al. (2014), Kalchbrenner & Blunsom (2013), and Bahdanau et al. (2015), can be considered as a kind of word sequence or byte sequence prediction task described in Ling et al. (2015). In the process of sequence prediction, NMT models predict the word based on the probability distribution of the target language's vocabulary. Therefore, for neural networks, we can combine each probability distribution to form an overall prediction task, thereby combining multiple models together.

For instance, Koehn (2017) exemplified a simple ensemble method to combine the probability distribution of candidate models. In decoding, each model predicts the next translated word and outputs a probability distribution. Here, the system can combine the probability distributions output by multiple models and average this combined distribution. The averaged probability distribution will be used to select the final output word. Further, Bojar et al. (2014) have tried to weight the output layer during prediction.

In addition to performing the ensemble during model decoding, Koehn (2017)

also mentioned a strategy that trains model decoding using opposite directions (right-to-left and left-to-right) and then ensembles the models of both directions. Usually, a model is decoded from left-to-right in the order of natural language[2], but this strategy requires a right to left decoding in addition to the normal left to right decoding. Models in both directions can provide a list of candidate translations. Comparing the translation given by models in both directions, the system could select the appropriate candidate translation.

Garmash & Monz (2016) observe and compare two methods for ensemble combination: a global weighting function and a context-dependent gating network. They concluded that the method using the context-dependent gating network[3] yielded better results. They then demonstrated the effectiveness of weighting methods on the German-English and French-English translation tasks. While the context-dependent gating network may in principle capture more differences in language phenomena, at the same time it may also increase the training difficulty.

4.3.2 Proposed Model Ensemble

In this thesis, we utilize a weighting strategy for ensemble decoding, described as Equation 4.2.

$$y = W_1\, y^1 + \ldots + W_m\, y^m \tag{4.2}$$

Here y^i refers to the translation probability distribution of the output layer of each candidate model, and w_m represents the weight of each candidate model. Generally, the more important the model, the more it needs to influence the translation result, and the higher its weight value.

This thesis tried two weighting strategies. The first weighting strategy is uniform, which means that the importance of each model is the same.

[2]In this thesis, we only consider languages in which a text is written left-to-right.

[3]Context-dependent gating network: While inputting training data, the gating network can also accept an additional reference input similar to the input to control training balance. In their study, the reference input was generated by context-dependent information.

The second weighting strategy is weighted. This strategy considers the importance of each model differently. The measure of importance is a ranking value related to the perplexity measure described in Section 4.2. The smaller the perplexity, the less difficult it is for the model to give the correct answer. It means that the model is more important.

The strategy of weighting is performed in the following way. All candidate models are sorted according to perplexity. In a specific case, for example, if the number of candidate models is three, the first-ranked model has a value of 3, the second-ranked model has a value of 2, and the third-ranked model has a value of 1, then the values are regularized to obtain the final weight w.

With regards to the comparison of the two weighting strategies of the model ensemble, there are some related studies reported in Sajjad et al. (2017). After comparing with the uniform strategy in Sajjad et al. (2017), it was found that the results of the weighted strategy are significantly higher than the uniform strategy by 1 or 2 BLEU scores. Although both the research in (Sajjad et al., 2017) and this thesis use the weighted strategy, the weighted strategy in Sajjad et al. (2017) is different from this thesis in that it weighs the scores over the vocabulary from each of the models. Returning to the two weighting strategies used in this thesis, let us look at the difference between the uniform and weighted strategy based on perplexity rank.

We briefly compare the effects between two kinds of weighting strategies in Tables 4.3 and 4.4. These tables list the BLEU values of running the model ensemble twice according to the uniform strategy or weighted strategy when the translation direction is ja-en or ja-zh.

The first run, Test 1, performed a model ensemble on five candidate models of the domain. The second run, Test 2, was performed to select five candidate models based on perplexity from all candidate models for ensemble decoding.

From the results we can see that, consistent with (Sajjad et al., 2017), the strategy (weighted) which weighs the importance of each model based on perplexity performs better than the other strategy (uniform). For instance, for the test set of

Table 4.3: Weighting strategy: uniform vs. weighted (ja-en)

	Test 1 (uniform)	Test 1 (weighted)	Test 2 (uniform)	Test 2 (weighted)
NEWS	41.64	42.87	40.11	43.09
Travel	30.34	32.78	28.25	30.57
IT	34.92	36.32	32.06	35.92

Table 4.4: Weighting strategy: uniform vs. weighted (ja-zh)

	Test 1 (uniform)	Test 1 (weighted)	Test 2 (uniform)	Test 2 (weighted)
NEWS	38.42	39.77	37.44	40.34
Travel	29.54	31.80	27.25	29.05
IT	33.91	36.68	34.52	35.17

the IT domain in Table 4.3, the BLUE score of Test 1 (weighted) 31.80 is better than that of Test 1 (uniform) 29.54. What is more, the BLUE score of Test 2 (weighted), 29.05, is also better than that of Test 2 (uniform), which is 27.25. Due to this result, we choose the default strategy of the model ensemble to be the weighted strategy unless it is clearly stated.

4.4 Chapter Summary

By utilizing model selection based on the perplexity measure in Section 4.2, we can determine the best models for the input sentence even without knowing the domain information of the input sentence. By adopting the approach of model ensemble in Section 4.3, we could combine the selected models to output the best translation result using the weighted strategy.

Chapter 5

Model Tuning

This chapter discusses the approaches related to model tuning. Using these approaches, we can optimize each domain-specific model, and finally improve the translation performance of the proposed multi-domain translation system.

In Section 5.1, we will briefly describe the four types of problems encountered by each model in actual translation tasks. These problems can be summarized as corpus quality, balance between in-domain and out-of-domain corpora, translation errors, and unknown words.

In Section 5.2, we will introduce four approaches to solving these problems, which can be summarized as adding noise, adjusting corpus ratio, replacing mistranslation, and utilizing subword.

5.1 Problems Description

For model tuning, we will consider and solve four specific problems.

5.1.1 Corpus Quality

In preliminary experiments, we defined a model trained with the corpus that contains bilingual text generated by an existing NMT model. For generating bilingual text, we translate the monolingual corpus by utilizing the existing NMT model. It could be found that the generated bilingual text sometimes reduces the

translation performance of the model.

It is easy to consider the quality of artificially generated sentences in the source language not as good as natural sentences. But through further analysis of more specific reasons, we can consider the key reason is that such artificial bilingual text lacks diversity.

More specifically, when generating sentences by using an existing NMT model, the algorithm of "Greedy search"[1] or "beam search" [2] will be used. These algorithms predict output based on MAP (Maximum A Posterior) estimation. MAP estimation searches and outputs only near the vertices of the distribution of the existing model. This means that these translated sentences created by MAP estimation are too easy to learn and do not provide sufficient training information. Since MAP estimation tends to provide a translation result with the highest probability, and often ignores the correct translation with a small probability (exception), these translations may be considered as low-diversity source sentences that do not adequately cover the text of the real world. This makes the translated artificial corpus seem to be insufficiently diverse in specific situations. The lack of diversity of the artificial corpus will often make the final model tend to output the translation which is too close to the output of the existed NMT model.

As an improvement scheme, we are inspired by the method proposed in Edunov et al. (2018) to add noise. The operation of adding noise could be considered as deleting or replacing a word with another one. By adding noise, a sentence can be expanded into multiple versions to increase the diversity. As a result, adding noise increases the difficulty of the model in learning the corpus

[1]Greedy search: When NMT outputs results, the system will choose the output words with the highest probability in the current state. In greedy search, this process is repeated until a terminator word is outputted.

[2]Beam search is a variant of greedy search, which usually sets a beam size n. We assume that n is equal to 2 and vocabulary consists of three words: a b c. When the NMT predicts the first word, the two words with the highest probability are selected to form two current output sequences, for example: "a" or "b". When predicting the second word, the system combines the current sequence "a" or "b" with all the words in vocabulary, respectively, to obtain the new six sequences "aa" "ab" "ac" "ba" "bb" "bc", and then selects the two with the highest probability from them as the current output sequence. This process is then repeated until a terminator word is outputted. Finally, two word sequences with the highest probability are outputted.

by slowing down the convergence speed of the training. By doing so, the model can better overcome the limitation of the initial (non-MT translated) corpus for training the existing NMT model.

5.1.2 Ratio of In-domain Corpus to Out-of-domain Corpus

Since in-domain corpora are more difficult to collect, the size of real in-domain corpora is mostly smaller than out-of-domain corpora. Although we can effectively use the in-domain and out-of-domain corpora to train a domain specific model while utilizing the domain adaptation method, the difference between the ratio of in-domain to out-of-domain data in the training data will also finally affect the translation performance of the domain specific model.

Therefore, how to adjust the size of in-domain and out-of-domain in the training data is also an important issue to be discussed.

5.1.3 Mistranslation

Mistranslation refers to the translation errors in translation result. This thesis focuses on two types of translation errors.

The first type of translation error is when a word of the input text is translated into a wrong word.

(1) a. SSML タグを使用して韻律を修正する
    b. Use SSML tags to modify traffic

Like Example (1), the sentence a refers to input text and the sentence b refers to the translation. We found that the Japanese word “韻律” which means “prosody” has been translated into the wrong word “traffic”.

The second type of translation error is inconsistent translation. Inconsistent translation means that there are multiple translations for the same word in the source language. When translating text of a specific domain, we always assume that the words translated will be standardized and uniform. If nouns such as technical terms are translated into different results, it will be considered a problem.

(2) a. 課金管理者および課金権限について詳しくは、アクセス制御の概要をご覧ください。

b. For more information about billing manager and billing permissions, see Overview of access control.

(3) a. 次の操作を実行する必要があるユーザーに、課金管理者の役割を割り当てます。

b. Assign the billing administrator role to users who need to do the following:

While observing Example (2) and (3), for the Japanese word“課金管理者”, the translation results in Example (2) and (3) are different. The result in Example (2) sentence b is“billing manager”, while that of the result in Example (3) sentence b is“billing administrator”.

For more detailed information, this Japanese word“課金管理者”is a compound word. The NMT model treats it as a combination of three words:“課金”“管理”“者”. In this case, the translation of these three words will be output separately, and the information referenced each time will be different according to the context of the current input sentence. Therefore, as shown in Example (2) and (3), “manager” is translated into two different results.

5.1.4 Unknown Words

The problem of unknown words is when the word in the input text has not been translated. That is, the original words were directly output into the results without translating.

(4) a. タッチパッドはタイピング時のホームポジションをできるだけ崩さないように、パームレストの中央、もしくはそのやや左側に配置されている。

b. The touchpad is located on the center of the パームレスト, or slightly on the left side, so that the touchpad can never die as possible.

An example of unknown words taken here is Example (4). Following Examples 1 to 3, sentence a refers to the input sentence in Japanese which is the source language, and sentence b refers to the translation result in English which is the target language. In the translation result, the Japanese word “パームレスト” is directly outputted into the English sentence without translating.

The main reason for the unknown words could be considered as a problem stemming from the small size of the model's vocabulary. As mentioned earlier in Chapter 1, NMT models inevitably limit the vocabulary size of the model because large vocabularies require a high computational cost. In order to ensure that the vocabulary of the model can cover the entire corpus to a large extent as possible, the words in vocabulary are usually selected according to the word frequency from high to low. As a result, this situation will cause the translation model to perform poorly when facing such low-frequency words which are not included in the vocabulary, and thus not translate them, instead outputting an unknown word.

In the scenario of multi-domain translation proposed in this thesis, we will perform model ensemble. Models included in the ensemble may correspond to different domains. When utilizing a model ensemble, the vocabulary of all models must be consistent. This means that the vocabulary cannot contain words characteristic for a domain but having low frequency. This exacerbates the problem of unknown words.

It is necessary to discuss a solution so that the vocabulary can cover all words in the corpus.

5.2 Problem Solution

In this section, we will elaborate four kinds of specific solutions to overcome the four types of problems described in Section 5.1: adding noise for the problem of corpus quality, adjusting the ratio for the problem of ratio of in-domain to out-of-domain corpora, replacing words and phrases for the problem of mistranslation, and utilizing subwords for the problem of unknown words.

5.2.1 Adding Noise

As mentioned in Section 5.1.1 earlier, due to MAP estimation, the translations produced by an existed NMT translation may always lean towards the optimal translation which lacks diversity. The lack of diversity will impact the training performance of the NMT model.

Edunov et al. (2018) made various comparisons during the translation process and found that the method of generating translations based on "beam+noise" is very effective in reducing the impact of the lack of diversity in the translation generated by the existing NMT model (see details in Figure 5.1).

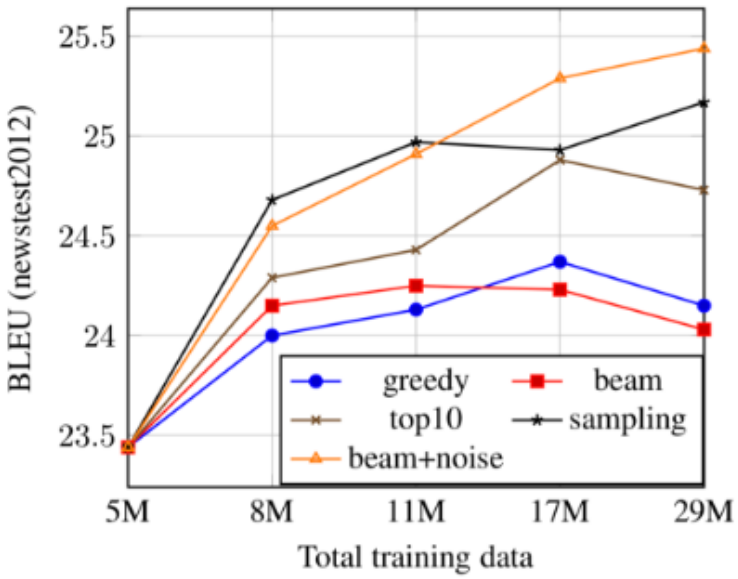


Figure 5.1: Training translated data
(adopted from Edunov et al. (2018), figure 1)

In Figure 5.1, there are several methods adopted in the translation process, and the BLEU result refers to the translation performance of the final NMT model trained using the artificial bilingual corpus generated by those methods. The "greedy" or "beam" here refers to different specific algorithms of MAP estimation

for exploring the output during the translation process.[3] The "sampling" refers to a method which generates outputs that are very diverse but sometimes highly unlikely. The "top10" refers to a type of restricted sampling. At each time step, it searches the 10 most likely words from the output distribution, renormalizes and then selects output from this sample set. The "beam+noise" means adding noise to the output translated using the "beam" method. From Figure 5.1, it can be clearly seen that "beam+noise" works best in all methods on the premise that the training data is greater than 10M. For instance, while the size of training data is 17M, "beam+noise" is 1 point higher than the BLEU score of "beam". At the same time, when the size of training data is 17M, "beam+noise" is still higher than the 2nd highest method "top10" by about 0.3 BLEU points.

Therefore, this thesis utilizes the scheme of adding noise with the best effect reported in Edunov et al. (2018) to improve the artificial translated corpus for training. We just apply an existing NMT to translate a monolingual corpus to a bilingual corpus, and then add noise to the bilingual corpus. The method proposed by Edunov et al. (2018) adds three types of noise.

- Delete a word with a probability of 0.1.
- Replace a word with a mask word with a probability of 0.1.
- Exchange the position[4] of words in the same output with a uniform distribution probability.

Compared with artificial sentences generated by MAP estimation, adding noise can make training a new translation model more difficult since it could be more diverse. Here, Edunov et al. (2018) cites examples of information perplexity to evaluate the diversity of artificial translated sentences. Generally, the higher the perplexity, the more diverse the sentences are.

In Figure 5.2, all methods of the generated translation are the same as described in Figure 5.1. Method "bitext" refers to training the model with real bilingual data

[3]See Section 5.1.1 for details.

[4]The distance between exchanged words should be limited to less than three positions.

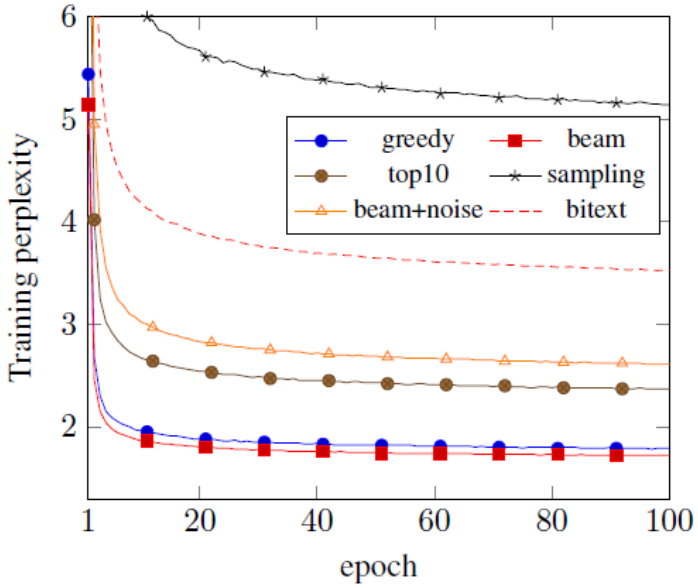


Figure 5.2: Perplexity of training translated data
(adopted from Edunov et al. (2018), figure 2)

using all generation methods, then averaging the perplexities over all generation methods. Epoch refers to the training steps. As the training steps increase, the complexity will gradually decrease. From Figure 5.2, we recognized that the " beam+noise " method has a higher perplexity than other generation methods, like the greedy and beam ones. Although the "sampling" method, which samples output from the model distribution, has the highest perplexity, we cannot adopt such a method. This is because the "sampling" method is even more difficult than the "bitext" method which is trained with real bilingual text.

5.2.2 Adjusting the Ratio of In-domain to Out-of-domain Data

When performing domain adaptation, we often use both in-domain data and out-of-domain data. Because of the difficulty of collecting data, we know that in-domain data is usually smaller than out-of-domain data. For instance, the ratios of in-domain data to out-of-domain data which is available for this thesis are always

smaller than 1:10, as detailed in Chapter 6.

In this case, it becomes key whether the model can learn parameters from two types of data in a balanced manner during training.

In simple terms, we can increase the smaller dataset to the same order of magnitude as the larger dataset by oversampling. Returning to the case of this thesis this means enlarging the in-domain data to the same size as the out-of-domain data. The method of oversampling is shown in Figure 5.3.

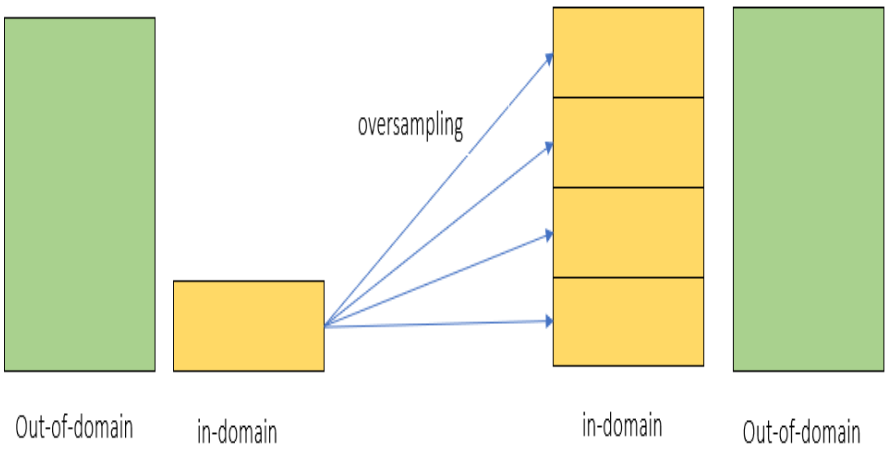


Figure 5.3: Oversampling

On the contrary, another method is undersampling, which reduces the larger dataset to the same level as a smaller dataset, as shown in Figure 5.4.

This thesis utilized the oversampling method for two reasons. The first one is that NMT learning requires a large amount of corpus data. As mentioned in Chapter 1, the training effect of NMT on small datasets may be worse than that of traditional SMT. If necessary, we avoid to train using a smaller corpus.

The second one is that it is difficult to extract a subset from out-of-domain data. Undersampling means that a great variety of out-of-domain data needs to

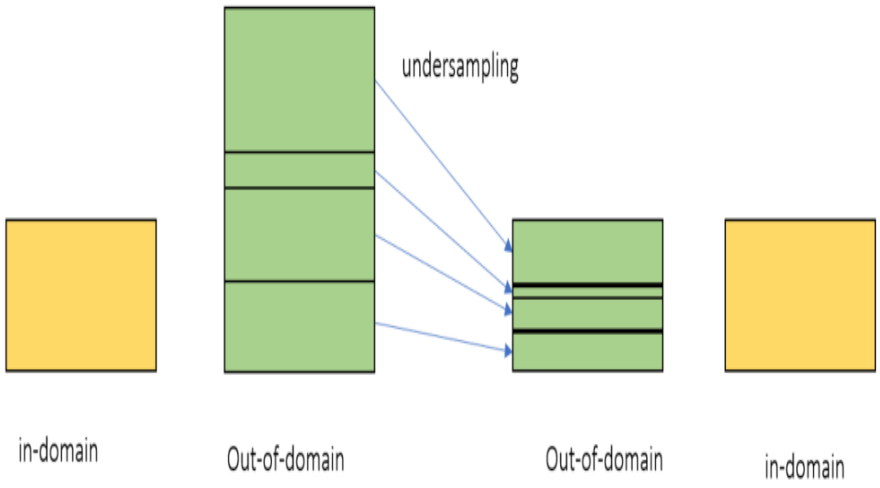


Figure 5.4: Undersampling

be excluded. However, there is no convenient method for judging data which is necessary. Just randomly sampling a subset from out-of-domain data is unrelated to whether the data is important for model training or not.

In the actual implementation of oversampling, we do not change the data of the in-domain data itself, only copy it to bring its amount close to the out-of-domain data. After oversampling, although a piece of the data in the in-domain data may actually the same as other pieces of data, later pre-experiments showed that oversampling would cause changes in training the model and improve the translation quality of translating in-domain text. One reason for this is that oversampling could increase the word frequency of in-domain. As described in (Koehn & Knowles, 2017), the model could perform poor while translating low-frequency word. For some important in-domain vocabularies, the model can cannot be learned them well since they are considered as low-frequency word of out-of-domain data while training the model. After oversampling the in-domain data, the NMT model may be more likely to capture the information of such words.

Table 5.1: Ratio adjustment (ja-en)

The Ratio of in-domain to out-of-domain, and the improvement.

	Ratio (News/out)	Improve	Ratio (Travel/out)	Improve	Ratio (IT/out)	Improve
ALL	1.0	0	1.0	0	1.0	0
fine-tuning	0.9	+0.89	0.9	+0.54	1.3	+1.20
mixed fine-tuning	1.1	+0.25	0.8	+1.30	1.0	0
stacking	1.0	0	0.7	+0.57	0.7	+0.34
data selection	3.0	+1.24	3.0	+1.06	2.0	+2.55

Table 5.2: Ratio adjustment (ja-zh)

The Ratio of in-domain to out-of-domain, and the improvement.

	Ratio (News/out)	Improve	Ratio (Travel/out)	Improve	Ratio (IT/out)	Improve
ALL	1.0	0	1.0	0	1.0	0
fine-tuning	0.9	+0.70	0.9	+0.28	1.2	+1.54
mixed fine-tuning	0.8	+1.62	0.8	+1.02	1.1	+0.98
stacking	1.0	0	1.0	0	1.0	0
data selection	2.0	+0.66	2.0	+0.38	2.0	+1.74

What is more, in actual experiments, the case setting the ratio of the two data sets (in-domain and out-of-domain) to 1:1 often could not bring out the best result. Therefore, we also consider a better setting for adjusting the ratio of in-domain to out-of-domain data as displayed in Tables 5.1 and 5.2. Tables 5.1 and 5.2 refer to training a model for Japanese-to-English (ja-en) or Japanese-to-Chinese (ja-zh), respectively.

The setting corresponds to each specific domain and specific domain adaptation methods. Specific domain adaptation methods refer to the five domain adaptation methods described in Chapter 3. The specific domains are of three types, here referred to as the News, Travel, and IT domain. More detailed discussion around the domain types will be conducted in Chapters 6 and 7. The “Ratio” means the ratio of in-domain data (specific domain) to out-of-domain data, and “Improve” means the improvement to the BLEU score after changing the setting from 1:1 to this ratio. For instance, in Table 5.1, a Ratio of 0.9 can lead to a 0.89 increment of BLEU score while training a News-specific model using the mixed fine-tuning method. Given one more example, in the case of ja-en in Table 5.1, when the method of fine-tuning is used, adjusting the ratio to 0.9 can improve the translation performance on the Travel domain by 0.54 BLEU points.

The Ratio usually refers to the ratio of in-domain data to out-of-domain data in the same training epoch, but some additional explanations follow. The fine-tuning and stacking methods do not use in-domain data and out-of-domain data together in the same training epoch. So the ratio according to these methods is actually the ratio of the in-domain to the out-of-domain subset between two different training epochs.

On the other hand, in the data selection method, unlike the other four domain adaptation methods, not all out-of-domain data is used. Instead, a part of the out-of-domain data is extracted by calculating semantic similarity, then trained together with the in-domain data. The ratio in the row of data selection is a ratio of the in-domain data to extracted out-of-domain data. For the experiment in Chapter 7, we select sentences from out-of-domain data which have alignment scores higher[5] than the threshold[6] and extract a corresponding amount[7] data from the out-of-domain data to expand the in-domain data (News, IT, Travel).

5.2.3 Replacing Words and Phrases

As discussed in Section 5.1, mistranslations or unknown words may still remain in the output result of NMT.

We consider applying replacement to process such errors. Replacement means that for some alternative content of the source language, we independently replace it with corresponding content of the target language by utilizing external knowledge and without relying on NMT. The alternative content could be a word or phrase.

The example of Word replacement could be considered as Example (5). The Japanese word “マッカーシー” could be replaced with the English word “Mc-

[5]An alignment score is a measure used to detect the similarity between sentences. For further details, see Section 3.3.5.

[6]For Japanese-to-English translation, the threshold is 0.505 for the News domain, 0.646 for the Travel domain, and 0.632 for the IT domain. For Japanese-to-Chinese translation, it is 0.467 for the News domain, 0.528 for the Travel domain, and 0.550 for the IT domain.

[7]For Japanese-to-English translation, we extract about 1.3 million, 0.7 million, and 0.7 million sentences for the News, Travel, and IT domains, respectively. For Japanese-to-Chinese translation, we extract about 1.5 million, 0.8 million, and 1 million sentences for the News, Travel, and IT domains, respectively.

Carthy".

(5) a. マッカーシーの 1958 年の論文では、2 つのタイプの表現が導入されている。

b. (before replacement) In the paper of マッカーシー, two types of representations have been introduced.

c. (after replacement) In the paper of McCarthy, two types of representations have been introduced.

On the other hand, the example of phrase replacement could be considered as Example (6). The Japanese phrase "ログオン認証パケット" was mistranslated as "ログオン logon authentication packets". Hence, the mistranslation can be replaced with "logon authentication packets".

(6) a. ログオン認証パケット の不達や遅延によるアクセス不能問題を起こしやすい。

b. (before replacement) It will easily cause inaccessibility problems due to missed or delayed ログオン logon authentication packets.

c. (after replacement) It will easily cause inaccessibility problems due to missed or delayed logon authentication packets

We could utilize a bilingual dictionary for replacement. Since we can obtain the alignment information between input text on the source language side and translation output on the target language side, for content in the input text, we also can replace related content in the translation output with a bilingual pair from a bilingual dictionary.

As the same idea, for phrase replacement, we need to utilize a bilingual phrase-pair list. This list can be obtained through SMT. For specific strategies, we adopted the scheme proposed by Long et al. (2019).

They first trained the text using SMT and got candidate phrase pairs. Then, they applied a statistical measure called branching entropy to extract phrase pairs from candidates phrase pairs. Branching entropy measures the probability that

a phrase belongs to a part of a longer phrase. If the phrase is a subphrase of another longer phrase, it possesses some adjacent phrase occurring at its left or right side. In this situation, there is a tendency that the value of branching entropy could be decreased. Conversely, if the phrase is a sub phrase of another longer phrase, the value of branching entropy could be increased. Therefore, we should extract the phrase whose value of branching entropy should be relatively large. The calculation of branching entropy is as the following equations 5.1 and 5.2.

$$H_l(t) = -\sum_{v \in V_l(t)} P_l(v|t) log_2 P_l(v|t) \tag{5.1}$$

$$H_r(t) = -\sum_{v \in V_r(t)} P_r(v|t) log_2 P_r(v|t) \tag{5.2}$$

In equations 5.1 and 5.2, $H_l(t)$ and $H_r(t)$ are left branching entropy and right branching entropy, respectively. t refers to the current phrase, and v represents all the phrases adjacent to t. Here the conditional probabilities of v and t are defined as $P_l(v|t)$ or $P_r(v|t)$ as in the following equations 5.3 and 5.4. $f(t)$ refers to the frequency of t, and $f(v,t)$ refers to the frequency of the occurrence both on v and t.

$$P_l(t) = \frac{f(v,t)}{f(t)} \tag{5.3}$$

$$P_r(t) = \frac{f(t,v)}{f(t)} \tag{5.4}$$

If the left branching entropy and right branching entropy of the phrase is over a threshold, the phrase t will be extracted. The threshold could be adjusted by the goal of collecting a certain amount of phrase pairs. As a result, we extracted and utilized about 25,000 and 13,000 phrase pairs for ja-en and ja-zh, respectively.

5.2.4 Utilizing Subwords

Section 5.1.4 discussed that the small size of the model's vocabulary could cause the problem of unknown words. An NMT model with a large vocabulary size

requires a related huge computational cost. This situation has forced NMT developers to extract a smaller vocabulary by ranking using word frequency. This will make an NMT model behave differently for low frequency words and high frequency words. In order to overcome the limitation of vocabulary size of the model, we could consider using subword encoding to improve this problem.

The principle of subwords is that if given a fixed vocabulary size, there is a vocabulary of (potentially) shorter word units which can cover all the units of the whole corpus. The idea of subwords is simple. In short, high-frequency words could be considered as one word, and low-frequency words could be divided into shorter units. The unit finally segmented in this way is called a subword, since it could be smaller than a word. In this way, the total number of "words" has been greatly reduced. This process of segmenting into subwords can be repeated until the total number of "words" becomes lower than the required vocabulary size.

Byte Pair Encoding (BPE) (described in more detail in (Sennrich et al., 2016c)) is a conventional subword segmentation algorithm that uses the text compression rate as an objective function to greedily determine the segmentation. BPE was originally applied in the field of data compression, and now it has also been widely used in the NMT field.

For this thesis we applied the toolkit SentencePiece presented by Kudo & Richardson (2018) for subword segmentation. The tool SentencePiece has made some optimizations and improvements over traditional subwords. Firstly, it improves the speed of the BPE algorithm, which is extremely critical for NMT tasks that often deal with large-scale text. Secondly, since Asian languages such as Chinese and Japanese do not have natural word boundaries, they often need to be segmented as a pre-process. SentencePiece has also adapted to this problem, so that subword segmentation can be performed without prior word segmentation.

(7) a. _タッチパッド _は _タイピング _時 _の _ホーム _ポジション _を _できる だけ _崩 さ _ない _よう _に _、 _パーム レスト _の _中央 _、 _も しくは _その _やや _左側 _に _配置 _さ _れ _て _いる _。

b. _The _touchpad _is _positioned _on _the _middle _of _Palm rest , _or

_slightly _left , _so _that _it _can ’ t _collapse _as _much _as _possible .

The example (7) here is the result of subword segmentation with SentencePiece. As describing about unknown words earlier, the Japanese word “パームレスト” is outputted by NMT without translation. The reason for this is that “パームレスト” is a low-frequency word which is not included in the model vocabulary, so the NMT model cannot translate it.

Since we cut the word into smaller pieces using SentencePiece, we need a symbol to mark where the real “beginning of the word” is. The underbar “_” in this example is used as the beginning mark. After subword segmentation, the units of “_パーム” and “レスト” in the example (7) sentence a are no longer “low-frequency words” for the NMT model. Then as the final English translation result in example (7) sentence b, “_パーム” and “レスト” are correctly translated into “_Palm” and “rest”.

5.3 Chapter Summary

In Section 5.1, we considered four types of problems which could influence each candidate model to translate the sentence well. These problems are listed as corpus quality, balance between in-domain and out-of-domain corpora, translation errors, and unknown words.

Correspondingly，we described four approaches in Section 5.2 to solve these four problems. The solution of adding noise was proposed to solve the problem of corpus quality. The solution of adjusting the corpus ratio was proposed for solving the problem of the ratio of in-domain corpus to out-of-domain corpus. The solution of replacing words and phrases deals with the problem of mistranslation. The solution of utilizing subwords corresponds to the problem of unknown words.

Chapter 6

Corpus Resources

In this chapter, we will discuss the corpus for training translation models.

A simple definition of a corpus is a language material for computer use and processing. Corpus is a large and structured set of texts. These materials can be news or novel, which is written by the language of our daily lives.

With the continuous development of computer technology, computers can process massive data quickly, then become an essential tool for utilizing corpus.

Hence it has given the start of utilizing corpora for machine translation. As a search area that is using computers to process text, machine translation has also started to make utilize corpus. For neural machine translation in this thesis, the main objective of utilizing a corpus is to train a translation model.

In Section 6.1, we consider how to collect corpus. The method of collecting corpus data depends on each corpus resources. We may collect corpus data from the prepared corpus, or some completely unprocessed corpus.

In Section 6.2, we further process the collected corpus so that it can meet the requirements for training translation models. The process of sentence alignment aims to make the bilingual corpus to be parallel (aligned) on sentence level. The process of data augmentation refers to the process to increase the size of training data. For instance, we consider to translate monolingual corpus to expand the bilingual corpus for training. The process of data filtering eliminates garbage sentences with poor quality, such as an English sentence occurred in Japanese

corpus.

In Section 6.3, finally we will describe the types and statistics of constructed corpus.

6.1 Data Collection

This thesis consider to utilize the corpus resource from two views, the existing corpora and texts acquired from Web sites. The existing corpora refers to corpus prepared with a defined structure. For web content, we applied web crawling to collect the web pages and web scraping to extract the text.

6.1.1 Utilizing Existing Corpus

In this thesis, many existing corpora were utilized. Some of the main corpora are briefly introduced below.

1. Donga corpus[1]

   Donga corpus is a multilingual parallel corpus. The Donga is a newspaper in the Republic of Korea. It was first published in 1920. In addition to Korean which language is the origin of the content, the newspaper also provides Japanese, Chinese and English versions of the same content as a paid service. The corpus is paralleled on document-aligned level.

2. ASPEC [2]

   The corpus of ASPEC (Asian Scientific Paper Excerpt Corpus) is a bilingual parallel corpus. It has already widely used in machine translation research. It mainly contains abstracts of scientific papers. The parallel language pairs are Japanese-to-English and Japanese-to-Chinese. The corpus is aligned to sentence level.

[1]Donga: http://www.donga.com/jp
[2]ASPEC: http://lotus.kuee.kyoto-u.ac.jp/ASPEC/

3. Opus [3]

   The corpora of OPUS (open source parallel corpus) focused on collecting bilingual translations from the Internet. Hence depending on the content, it could provide many different language pairs. All the texts is preprocessed automatically by tool, and files are provided in multiple formats. The corpus is basically sentence-level aligned. We can collect many corpora from different genres.

4. United Nations Parallel Corpus [4]

   The United Nations parallel corpus is mainly composed of official records of United Nations conference documents and other parliamentary documents. Most documents correspond to the six official languages of the United Nations, namely Chinese, English, French, Russian, Spanish, Arabic. Unfortunately, this corpus does not have a Japanese version. According to the task settings detailed in Chapter 7, our translation tasks involve Japanese. So we will utilize the Chinese and English data as monolingual corpus respectively.

5. Wikipedia [5]

   Wikipedia is an online encyclopedia. It contains various terms and explanations of the terms. The corpus is very high in quantity and contains hundreds of millions of sentences. However, the articles under the same entry in each language do not have an absolute correspondence. In this thesis, we used the text extracted from Wikipedia data as a monolingual corpus.

6.1.2 Web Scraping

Web scraping is, in simple terms, a technique for obtaining text from a web page. The biggest feature of web content is that it is free and the amount of corpus is large

[3]Opus: http://opus.nlpl.eu/index.php
[4]UN Nations Parallel Corpus: https://conferences.unite.un.org/UNCorpus/
[5]Wikipedia (en): https://dumps.wikimedia.org/enwiki/
Wikipedia (zh): https://dumps.wikimedia.org/zhwiki/
Wikipedia (ja): https://dumps.wikimedia.org/jawiki/
Notice that, the Wikipedia is constantly updated. We utilized the latest version in Feb, 2019.

enough. The corpus obtained through web scraping can be used as a supplement to the existing corpus.

The web crawling strategy adopted in this research is to collect web pages from website. It start crawling pages from the top page of website, automatically explore downwards, and get all available pages (html files). Then after utilizing web scraping to parse the structured data in the html file, we can get the text of crawled web pages.

What is more, through web scraping we don't just get a great variety of monolingual content, but also the bilingual content. Since many website has multiple language versions, if we can find that there is a connection between web pages in different language, then at the same time we can obtain the corresponding multilingual content during scraping monolingual web contents.

6.2 Data Processing

The primary purpose of utilizing corpus is to train NMT model. Training NMT model requires bilingual corpus which is aligned on sentence level. But not all the corpus collected in Section 6.1 meet such requirement.

We will adopt three types of processing: sentence alignment, data augmentation, and data filtering. Sentence Alignment is to ensure that the data is aligned on sentence level. Data augmentations is to expand data from monolingual corpus to bilingual corpus. Data filtering is to clean up problematic sentence pairs in bilingual corpora, such as an English sentence occurred in Japanese corpus. These sentence pairs will reduce the training effect of the model.

6.2.1 Sentence Alignment

Depending on utilizing existed corpus and scraping web context in Section 6.1, we were able to obtain bilingual corpora. But some of the corpora cannot be directly used for model training, since they are not parallel corpora in sentence level. For example, the Donga corpus mentioned in section 6.1 are aligned on document

level, not sentence level.

Since when training the NMT model, the input samples that need to be input are sentences, for the collected bilingual corpora, we need to align such corpus on a smaller level (sentence) than the document level.

As the name implies, sentence alignment task is to compare the information between sentences and confirm whether the sentences are semantically corresponding. Specifically in a bilingual environment, alignment will measure whether a pair of sentences possess a considerable semantic similarity.

We used a simple method to achieve sentence alignment. That is to translate bilingual sentences into the same language, and then compare their similarity to determine whether sentences should be aligned.

For the convenience of description, after translating text into the same language, we would refer the text of both sides as text a and b. We refer sentences in text a as sentence a, sentences in text b and sentence b. The process of sentence alignment could be divided as following six steps:

1. Translation

   We will firstly translate sentences from one language to another language [6]. This idea of using translation will also be applied in subsequent Section 6.2.2 to expand the bilingual corpora.

2. Postagging

   We then utilize a parser to obtain the word segments of sentences and obtain the part-of-speech tag (postag) of the word. [7]

3. Determining a window

   We use the window method to select sentences as the alignment candidates on both sides of sentence. We consider the text is the sequence of sentences. Starting from the starting position of the text a, we firstly collect a range of

[6] Translation from English or Chinese to Japanese.

[7] Since the language of sentences is Japanese, we applied a Japanese parser Mecab. http://taku910.github.io/mecab/

sentences, and the range size are set as five. After determining the scope of sentences in text a, start from the starting position of text b to determine the scope of sentences in text b. The range size used in text b is 5 multiplied by a ratio which is equal to the ratio of the length of text a to text b. For example, if length of text b is twice as text a, then the range size used in text b is 10. (5 multiply by 2)

4. Calculating similarity

   For a specific window of selected alignment candidates, we calculate the similarity between the alignment candidates on both sides. In this window, the alignment candidate can correspond to multiple sentences. Since text pair is already aligned on text level, it promised that there must be related parts which should be aligned on both text. We do not need a particularly strict calculation to measure semantic similarity. Therefore, we utilized the simplest measure of similarity. That is, compare how many words on both sides are consistent. Similarity is calculated as the number of words which occur on both sides divided by the total number of words averaged over both sides. At the same time, using the postag information of the word, we found that comparing the words of only a few postags makes the alignment accuracy better, such as noun, adj, adv.

5. Output aligned sentences

   Then we output the sentence candidates with the largest similarity in the current window. If similarity measure of the candidate exceeding the threshold value, it could be considered as the aligned sentences. According to the actual processing experience, when the threshold is set to 0.2, ignoring the case of one sentence aligned multiple sentences, we extracted about 70 percent of Japanese-to-English sentence pairs from the original text, and about 65 percent of Japanese-to-Chinese sentence pairs from the original text.

6. Repeat align process

   If not found aligned sentences, add 1 to the starting position of the text a and b.

In the contrary, set the starting position depend on the last aligned sentences. Then starting from the remaining sentences after alignment, continue to repeat the above steps: determining a window, calculating similarity, output aligned sentences.

6.2.2 Data Augmentation

Data Augmentation refers to the process that creates artificial data from existing data to increase the amount of data. For this thesis, considering the possible shortage of bilingual corpora, we also collected monolingual corpora and expected to expand them into bilingual corpora. This process can also be regarded as a process of data augmentation.

In the studies of neural machine translation, researchers have performed extensive work to improve the use of monolingual data, including language model fusion proposed by Gülçehre et al. (2015), back-translation proposed by Sennrich et al. (2015) and dual Learning proposed by Xia et al. (2016), Cheng et al. (2016).

During the execution of these methods, Back-translation ((Sennrich et al., 2015)) is one of the most notable technologies. In this thesis, the term back-translation is used when we translate a corpus in the currently considered target language into the source language. That is, the translation direction is the opposite of the considered translation direction. Note that this definition of back-translation was proposed in (Sennrich et al., 2015), which may be different from another definition/usage of this term, that is: translating a once-translated sentence in language Y back into language X, which is the original language of the sentence.

We use back-translation to translate monolingual corpus here as the practical method to solve the data augmentation problem. Since it is less difficult to obtain a single language than a bilingual corpus, using back-translation could lead us to obtain more bilingual corpus. Back-translation is simple and easy to apply because it does not require to modify an additional language model or apply a specialized machine learning scheme. In fact, for workshops of evaluating machine translation, the organizer usually also provides a large number of monolingual corpora for

participants as training data.

Many studies have proven this method useful for SMT as Bojar & Tamchyna (2011), NMT as Sennrich et al. (2015) and Poncelas et al. (2018), both unsupervised SMT and NMT as Lample et al. (2018).

Specially for NMT, research of Edunov et al. (2018) shows that back-translation can be performed well for training machine translation model. In the report by Edunov et al. (2018), back-translation was set up and run in a semi-supervised learning environment for the technology. In their report, they claimed that the final result reached an outstanding score of 35 points on BLEU on the English-German test set of WMT ' 14.

For this thesis, the process of utilizing back-translation to train NMT model will be executed in the following order.

1. Confirming the machine translation results for back-translation
   Performing back-translation requires a back-translation model from target language to source language. In related works, like Edunov et al. (2018), they train such a model by themselves.

   In the practical application of this thesis, we have adopted existing translation systems to perform back-translation. Instead of back-translation models using real bilingual corpora, we make direct use of two commercial services offering machine translation systems: GTcom [8] and Google[9].

2. Construct artificial bilingual corpus
   Then we could generate the translation and combine these artificial sentences into the source language with the original sentences in the target language to form the artificial bilingual corpus.

3. Train the model

[8]GTcom: http://www.gtcom.com.cn/

[9]Google: https://cloud.google.com/translate/docs?hl=ja.

Notice that the commercial machine translation services are constantly updated. We utilized these machine translation services for expanding the corpus in July and August, 2019.

When training the NMT Model, the artificial bilingual corpus and the real bilingual corpus are trained together.

6.2.3 Data Filtering

We process sentence alignment for corpora aligned at the text level in Section 6.2.1, and process back-translation for the monolingual corpus in Section 6.2.2. Together, we could construct a sentence-aligned bilingual corpus from each corpus resource. However, there still remain a certain amount of bad sentence pairs containing bad content originating from the data collection phase or from the data generated during back-translation. These kinds of sentence pairs will impact the NMT model training process.

Data filtering, as applied here, is a process to simply exclude such bad sentence pairs from the bilingual corpus. In this thesis, a rule-based method is used to detect whether the actual bilingual sentence pair contains the bad content, and thus whether it should be excluded.

Considering that the source of the original corpus is complex and the types are different, we need to adopt a general and simple strategy which could treat different corpus resources at the same time. For a corpus that is originally a real bilingual corpus, because alignment has been performed before, the sentences of the corpus are aligned at the sentence level. In this way, the sentence in source language and its corresponding sentence in target language are guaranteed to be consistent in content. For a corpus that is originally a real monolingual corpus, we used the back-translation method to obtain bilingual sentence pairs. In this sense, the content between the sentence pair is also consistent.

Therefore, for the sentence pair, we no longer focus on the content consistency issue in this section, so the elements we need to detect are greatly reduced. In the actual work of this thesis, we mainly focus on checking the bad content in sentence pairs that could be listed under the following categories of errors.

1. Sentence pairs which have an empty Japanese sentence.

Bilingual sentence pairs obtained through back-translation are naturally considered to be aligned on sentence level, so we will no longer align such sentences. However, when processing back-translation by utilizing the machine translation service GTcom, if the input sentence is just a sequence of special symbols, there is a very low probability for the MT to output a blank sentence.

2. Sentence pairs for which the length of the sentence in the target language is 4 times longer than in the source language.

   This is in order to detect a common mistake generated during NMT translation: repeated translation of words. This phenomenon can be considered as some words being repeatedly output in the process of applying back-translation. It is better to exclude such sentences from training.

3. Sentence pairs in which a sentence on one side is too short.
   It is possible that short sentences will affect the training of the model negatively. We will check the length of the sentence pairs. If there are sentences with less than 3 words, we exclude such sentence pairs.

4. Sentence pairs in which one side contains multiple sentences.
   If periods appear at non-terminal positions, it means the sentence may contain multiple sentences. Such instances are not considered to form a good sentence pair and should be excluded from the training corpus. The main reasons these types of pairs appear is that, in addition to the poor quality of the original text, they may be caused by poor preprocessing, specifically during sentence segmentation.

5. Sentence pairs in which no terminator appears at the end of the sentence.
   Terminators here refer to periods, exclamation points, question marks, and quotes, that mark the end of a sentence. This phenomenon is also related to sentence segmentation.

6. Sentence pairs in which special characters appear at the starting position.
   Special characters refers to a character set that does not contain the English alphabet, Japanese Kana or Chinese characters. We do not consider sentences meeting this condition to be suitable, and excluded them from the training corpus. One reason for their appearance in the corpus is due to data collection issues that picked up sentences of poor quality. Another reason is that they could be introduced during the back-translation process if the model made an error when translating sentences into the target language, resulting in an unreasonable sentence structure in the translation.

7. Sentence pairs which lack characters corresponding to the language.
   For example, if there are no Hiragana characters in Japanese sentences, or no Chinese characters in Chinese sentences. Naturally, such sentences need not be included in the training data. This phenomenon in the case of Japanese sentences more commonly originates in back-translation failures.

6.3 Classification and Statistics of Corpora Used

In this section, we will classify the corpora into different types, and simply describe their statistics in terms of size and composition.

6.3.1 Classification of Corpora

For the evaluation task detailed in Chapter 7, we need to pay attention to the following three classification perspectives: language, translation direction and domain information.

1. by language

   From the perspective of language, the corpus could be divided into monolingual, bilingual and multilingual corpora. Furthermore, bilingual and multilingual corpora can be divided into parallel (aligned) corpora and non-parallel according to the organizational form of the corpus. This thesis firstly

focuses on the bilingual parallel corpora, since they are available for training the NMT model. As a supplementary resource to the bilingual corpus, we have paid attention to utilizing the monolingual corpus. Although in this thesis, all the corpora could be organized and transformed into parallel bilingual corpora. The information on utilizing the monolingual corpus will be described in Section 6.3.2.

2. by translation direction

   Translation direction refers to the direction of source language to target language. In this thesis, the translation direction was defined as Japanese-to-English and Japanese-to-Chinese, which will be detailed in Chapter 7. Therefore, for back-translation to transform monolingual data into bilingual data, it means to translate sentences of English or Chinese into Japanese.

3. by domain information

   Since we proposed a NMT framework for multi-domain translation in this thesis, we also need to distinguish the domains of the corpus. Clear domain information can guide us to construct the in-domain corpus.

   More specifically, we mainly make domain distinctions relying on the domain information of the corpus source. According to the consideration detailed in Chapter 7, if the corpus resource could be classified as one of the three domains (News, Travel, IT), it will be independently utilized as a domain-specific corpus, namely a News/Travel/IT corpus. The three domain-specific corpora will be utilized as in-domain corpora individually (see Tables 6.1 or 6.2). On the other hand, we built a General corpus[10], as seen in Table 6.1. The domain distinction was processed during the phase of collecting the corpus manually.

[10]General corpus contains all data we could collect, except the data from IT, Travel, and News domains. More specifically, we classified the collected data into IT, Travel, News, Financial, Legal, Medical and Other. In other words, the sentences from Financial, Legal, Medical and Other domains are collected in the General corpus.

Table 6.1: Size of Japanese-to-English bilingual corpus in millions of sentence pairs

	real	back-translation	total
News	3M	1M	4M
Travel	1.8M	0.2M	2.0M
IT	2.2M	0	2.2M
General	8M	100M	108M

Table 6.2: Size of Japanese-to-Chinese bilingual corpus in millions of sentence pairs

	real	back-translation	total
News	3M	1M	4M
Travel	1.6M	0.4M	2M
IT	2M	0	2M
General	3M	28M	31M

For example, if the corpus comes from the documentation of the programming language Python, then this corpus can be classified as a corpus in the IT domain.

6.3.2 Corpus Statistics

Through the processing described in section 6.2, we have obtained an aligned bilingual corpus of sufficient size. The bilingual corpus is constructed in Japanese-English and Japanese-Chinese to meet the need of the evaluation task in Chapter 7. Then according to the evaluation task of Chapter 7, we also categorize the corpus into the three domain types (in-domain: News, Travel, IT) and a General corpus (out-of-domain). The statistics of the corpora available to us is provided by Tables 6.1 and 6.2.

In Tables 6.1 and 6.2, Bilingual data (real) refers to the corpora obtained from existing bilingual corpora. The M in Tables 6.1 and 6.2 means 1 million sentence pairs. Bilingual data (back-translation) refers to the artificial bilingual corpora obtained through back-translation. Total is the sum of two types of bilingual data. What is more, through data filtering and actual training, we found that the quality

of artificial bilingual corpus on IT domain is not good for training model, so we did not use any back-translated corpus based on IT domain.

From Tables 6.1 and 6.2, we recognize the following situations. For General corpus, the corpus generated by back-translation is its main source. For the corpus of each in-domain, the corpus generated by back-translation is relatively small in number. On the other hand, General (out-of-domain) corpus contained more bilingual data (back-translation) than bilingual data (real).

For example, in Table 6.1 for Japanese–to-English corpus, the size of total General corpus, is 108M , which is lager than other in-domain corpora. In order to make an effective domain adaptation, the size of bilingual data (back-translation) on each in-domain is 1M, 0.2M, 0. Its are smaller than the size of bilingual data (real), 3M, 1.8M 2.2M. On the contrary, for General corpus, the size of bilingual data (back-translation) 100M is larger than the size of bilingual data (real) 8M.

The tendency presented by in Table 6.2 for Japanese-to-Chinese corpus is basically similar with Table 6.1. The size of total General corpus,is 31M, which is lager than other in-domain corpora. The size of bilingual data (back-translation)on each in-domain is 1M, 0.4M, 0, which are smaller than the size of bilingual data (real), 3M, 1.6M, 2M. For General corpus, the size of bilingual data (back-translation) 28M is larger than the size of bilingual data(real) 3M.

6.4 Chapter Summary

This chapter described the situation of preparing and constructing the training corpus. The corpus could be collected from existed corpora and web contents. Further, since training NMT model required bilingual aligned sentence pair, we utilized sentence alignment to align texts on sentence-level, and applied existing machine translation to convert the monolingual corpus into a bilingual corpus. We also eliminate poor quality garbage sentence pairs. Finally, we discussed the statistics and classification of training corpus.

Chapter 7

Evaluation

This chapter first describes how to evaluate the effectiveness of the multi-domain NMT system based on the proposed framework, and then discusses the evaluation results quantitatively and qualitatively.

In Section 7.1, we define the evaluation tasks and prepare the test sets related to these tasks. In Section 7.2, we will introduce two evaluation methods to evaluate the translation results on each task: Automatic evaluation and Human evaluation. In Sections 7.3 and 7.4, we will analyze the results of the evaluation. Finally, in Section 7.5, we will summarize the content of this chapter, give a brief conclusion based on the results of the evaluation, and discuss possible topics for future study.

7.1 Evaluation Task and Test Set

7.1.1 Evaluation Tasks

In this section, we define the evaluation tasks to evaluate the multi-domain translation system. We defined the evaluation tasks by considering three aspects: translation direction, domain type, and translation type.

Translation Direction

Translation direction determines the languages of input and output; in other words, the source language and target language. By considering potential transla-

tion needs, we define two translation directions: Japanese-to-English (ja-en) and Japanese-to-Chinese (ja-zh).

As described in the corpus preparation in Chapter 6, we will enable the back-translation[1] method to expand the monolingual corpus into a parallel corpus. However, back-translation inevitably bring noise, and the deviation caused by the noise will affect the effectiveness of the translation model.

In order to obtain the best translation performance, the text in the target language must consist of reliable sentences obtainable without back-translating.

Compared to Chinese and English, Japanese has fewer resources before back-translating. Hence, we define Chinese and English as target languages.

Domain Type

Domain type represents the type of domain to which the test set belongs. For evaluating multi-domain translation, the system should perform well in each domain of interest. In order to prove the effectiveness of the multi-domain translation system, we will evaluate the system for translating the texts of three domains: News, Travel, IT. Text not belonging to the three domains will be utilized as general text in this thesis.

1. News

   News mainly refers to news reports. News text, in a broad sense, includes text from news reports or news communication. News text has a relatively strict sentence structure and is fastidious about using written language. Example (8) presents an instance of News text.

   (8) a. 労働部長官は、この日決まった同案に対する異議の申し立て期間を経て、8月5日までに来年度の最低賃金を確定・告示する。
   b. Labor Minister Lee Young-hee will authorize the minimum wage for next year and announce it Aug.5 after a period for receiving complaints about the wage.

[1]As remarked in Chapter 6, we use the term back-translation as defined by Sennrich et al. (2015).

c. 另外，劳动部部长将在 对当天决定的异议申 请期限之后，至 8 月 5 日为止，确定、公布最低工 资。

2. Travel

   In this thesis, we consider Travel text as text which targets tourists seeking travel-related information. Travel text tends to use spoken language more than written language, and is usually shorter and more straightforward than News text. Travel text could be short and straightforward sentences. Example (9) is an instance of Travel text, whose syntactic structure is a noun phrase rather than a sentence.

   (9) a. 名高き日本の海に浮かぶ神社

   b. Japan's famous shrine is built over water

3. IT

   Texts in the IT domain mainly come from software documentation, which possesses a relatively professional style and contains content differing from the other domain text. Besides, as an essential feature, IT texts contain a large number of technical jargons that do not appear in other domains.

   In Japanese and Chinese texts, these kind of specialized vocabulary will also appear as-is directly in English. Since Japanese and Chinese IT texts included relatively many English jargons than in other domains, we consider that the IT domain relatively from non-IT domains to some extent. Example (10) shows an instance of this phenomenon.

   (10) a. 問題:Autodesk®Fabrication データベースからデータをエクスポートし、別の いずれかにデータをインポートする必要があります。

   b. 问题 : 需要从 Autodesk®Fabrication 数据 库导出数据并将数据导入 另一个数据。

Translation Type

The translation type designates the translation method used by the multi-domain NMT system. The translation type defines the range of candidate models[2] and the number of chosen models in the model selection. In Chapter 4, we proposed approaches to model selection and model ensemble, in that the models selected that are assessed as good models will be afterward ensembled to achieve better translation.

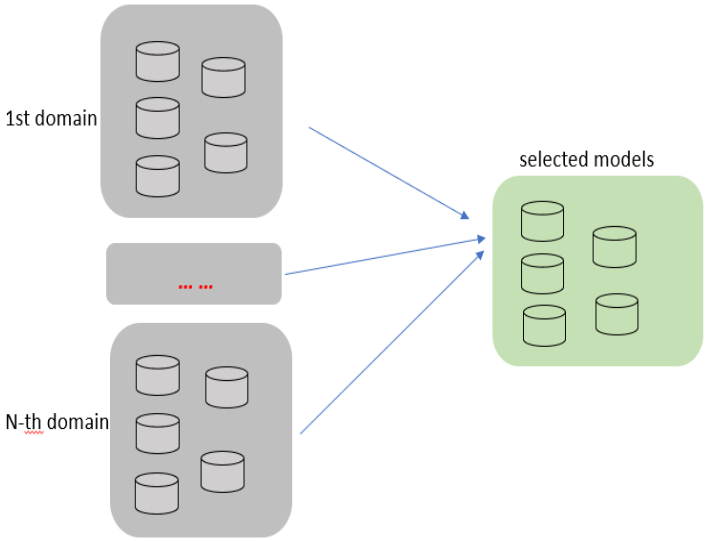


Figure 7.1: Translation type: multi_5

We define the translation type according to the specific translation method based on this translation approach. The translation type will specify the types of the candidate models for model selection. They are multi_n , domain_n, and baseline.

1. multi_n

   When we do not know the domain information of the input text, we select

[2]Candidate models of a specific domain or all candidate models.

models from all candidate models. This kind of translation type will be referred to as multi_n. The "n" will specify the number of models selected during model selection. For instance, if 5 models are selected, the translation type is multi_5 (Figure 7.1). If 3 models are selected, the translation type is multi_3. We define multi_5 as the basic translation type of the proposed multi-domain translation system.

2. domain_n:

   On the other hand, when we know the domain information of the input text, the range of the candidate models can be limited to the domain-specific models corresponding to the domain. This kind of translation type will be referred to as domain_n, as shown in Figure 7.2, where the first domain is chosen and 5 models are selected from this domain.

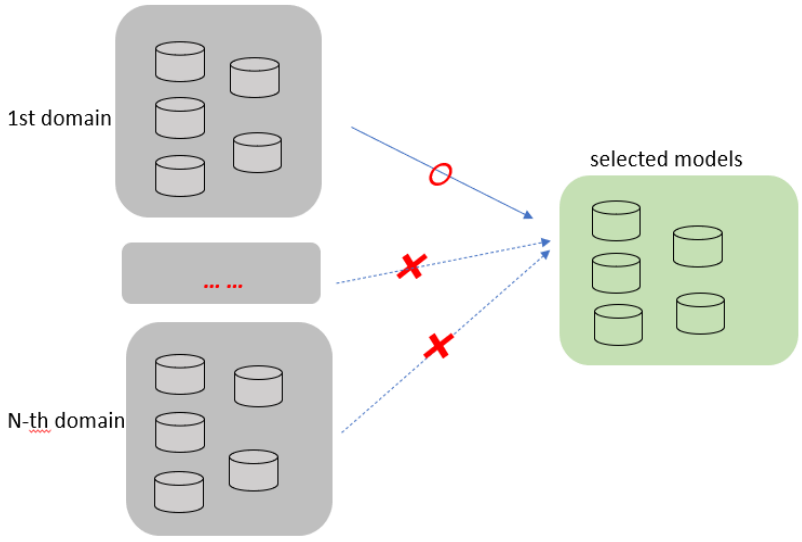


Figure 7.2: Translation type: domain_5

3. baseline

   Finally, when the translation type is baseline, it means that the model used in the translation process is the baseline model, which is trained on a corpus

simply combined from all available corpora (News, Travel, IT, and General). For training the baseline model, we did not use any of the approaches proposed in this thesis.

We define a specific evaluation task on three aspects: translation direction, domain type, and translation type. For example, a specific evaluation task could evaluate the Japanese-to-Chinese translation on the IT domain using the multi_5 translation type.

7.1.2 Test Set

The test set contains bilingual pairs of sentences for evaluation, which means the sentences are not used in the translation model training. The sentences of the source language refer to the input sentences in the evaluation task. The ground-truth sentences of the target language refer to the reference sentences for comparison with the system translation output.

As mentioned in Section 7.1.1, the translation directions of the evaluation task for this thesis are Japanese-to-English and Japanese-to-Chinese. We will prepare the same number of test sets for each translation direction. For instance, we prepare the IT test set of 2,000 sentence pairs for each specific translation direction individually.

In Section 6.3, we described in-domain (News, Travel, IT) and out-of-domain (general) corpus types. As shown in Figure 7.3, we extracted five test sets from the in-domain corpus and out-of-domain corpus: News, Travel, IT, Mixed and General corpus.

1. Specific domain test set (News, Travel, IT)

   We prepare the corresponding test set for the domain type, which corresponds to the second aspect of defining the evaluation task. In the description of the domain type, three specific domains with various styles and content are specified: News, Travel, IT domains.

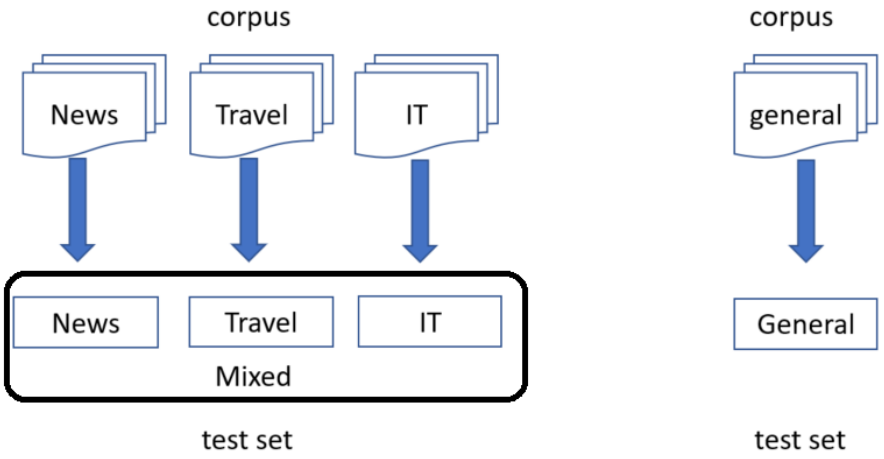


Figure 7.3: Test set: extracted from corpus

Therefore, we need to prepare three corresponding test sets for these three domains. We extracted three specific domain test sets from the in-domain corpus: News, Travel, and IT test sets.

For each specific translation direction, we choose 2,000 sentence pairs from the IT corpus as the IT test set, 2,000 sentence pairs from the Travel corpus as the Travel test set, and 2,000 sentence pairs from the News corpus as the News test set.

2. Mixed domain test set

   On the other hand, one of the purposes of this thesis is to provide a suitable translation result using the domain-specific translation model under the premise that the domain information of the input text is unknown. As a supplemental test set, we prepared a mixed domain test set. We mixed the three specific domain test sets (IT, Travel, News) into the mixed test set. Since the mixed test set contains sentences from three domains, it can better simulate a situation in which we do not know the domain information from the test set. For the translation type, we use the multi_n translation type to meet the situation of the mixed test set.

We combined the three specific domain test sets (IT, Travel, News) to form a mixed domain test set of 6,000 sentence pairs. Using the mixed test set, we are able to better analyze the effects of the model selection and model ensemble methods when faced with a mixed test set containing multiple domains.

3. General test set

   Although the translation model in our proposed translation system is specialized for specific domains, we also need to verify that our proposed translation system will not cause nasty effects when translating text which does not belong to the specific domain. If the translation types used in this multi-domain translation system also have excellent performance in translating out-of-domain text, such a multi-domain translation system can have more extensive adaptability to different domains.

   Therefore, for this purpose, we prepared a General test set. The sentences in this test set do not belong to the three specific domains (IT, Travel, News). We selected 3,000 sentence pairs from the General corpus as the General test set.

The corpus we prepared contains real[3] bilingual corpora and a large number of monolingual corpora. For training translation models, we used back-translation to convert these monolingual corpora into bilingual corpora. While extracting the test set from the corpus, we will avoid these translated bilingual corpora and only utilize the real bilingual corpora.

In order to avoid extracting too many sentences from the same original data, which biases the evaluation results, we shuffle the corpora, and then extract sentences from them. In addition, for one test set, we extracted at least 5 sentence pairs from each existing corpus described in Section 6.1.1.

[3]Real bilingual corpora are limited to directly collected bilingual data and do not contain synthetic bilingual data which was machine-translated from monolingual data.

7.2 Evaluation Methodologies

For evaluating, we adopted two evaluation methods. One is automatic evaluation, and the other is human evaluation. The characteristics of the two evaluation methods are different.

The most significant difference is the speed and ease of obtaining evaluation results. Automatic evaluation, as its name implies, obtains evaluation results at the speed of the evaluation algorithm, which is only constrained by available computing power. This differs greatly from human evaluation, with its speed constrained by the speed of the evaluator and their total capacity for work over a certain period of time. As such, while automatic evaluation could use the test sets prepared in Section 7.1, because of the greater difficulty and cost of human evaluation, the actual test set used for human evaluation is a smaller subset containing 100 sentences randomly selected from the test set used for automatic evaluation.

7.2.1 Automatic Evaluation

In order to measure translation performance for the automatic evaluation, we utilize the BLEU score proposed by Papineni et al. (2002). The BLEU score is a measure representing the similarity (distance) between a translation result and its reference sentence. A higher BLEU score can be restated as a shorter distance, meaning that the translation result is closer to the reference sentence. If the BLEU score of the translation result of a task is higher than that of another task, it means that the performance of this task is better than the other one. For our evaluation, the reference sentence represents the sentence of the target language in the test set. These correspond to the Chinese and English sentences.

The critical point of BLEU is its criterion for measuring the distance between machine translation results and reference sentences. By using a modified n-gram measure, BLEU defines a type of distance between the machine translation result and the reference sentence. This is called the modified n-gram precision, a worked-

through example of which is given in example 11 below.

(11) a. The cat is on the mat

b. (Machine-translated sentence) The the the the the the the.

c. (Reference sentence) The cat is on the mat.

For example, if the specified length of the n-gram is 2, the set of n-grams of sentence-a in example (11)[4] could be: "The cat", "cat is", "is on", "on the", and "the mat". The size of the n-gram set is 5. For BLEU, four types of n-gram lengths are defined (1, 2, 3, and 4 words for one segment part).

$$P_n = \frac{\sum_i \sum_k min(h_k(c_i), max_{j \in m} h_k(s_{ij}))}{\sum_i \sum_k (h_k(c_i))} \tag{7.1}$$

Modified n-gram precision P_n represents the distance between the machine translation result and the reference sentence and is calculated according to Equation 7.1.

Here, $\sum_i$ calculates the sum of the scores over the set of translation results, $\sum_k$ sums the scores over n-gram from n equals 1 through 4, the set of n-grams in one translation result, $h_k(c_i)$ is the count of the k-th n-gram in the i-th translation result, and $h_k(s_{ij})$ is the count of the k-th n-gram in both the i-th translation result and the j-th reference sentence. Finally, m means there are m reference sentences. For our actual evaluation, we have only one reference sentence in the test set. Hence m is always equal to 1.

For example, we compute the n-gram (unigram) precision P_n between sentence-b and sentence-c in Example (11) as shown in Equation 7.2. For this example, the size of the set of translation results $\sum_i$ is 1. The size of the set of the n-gram in one translation result $\sum_k$ is 1. "The" appears seven times in sentence-b and two times in the sentence-c. Since sentence-b possesses only one unigram "the", the count of $h_k(c_i)$ is 7. The count $h_k(s_{ij})$ is 2.

[4]We quote this example sentence from the following Wikipedia article: `https://en.wikipedia.org/wiki/BLEU`.

$$P_n = \frac{min(h_k(c_i), h_k(s_{ij}))}{(h_k(c_i))} = \frac{min(7,2)}{7} = \frac{2}{7} \tag{7.2}$$

$$BP = \begin{cases} 1 & if \;\; l_c > l_s \\ e^{1-\frac{l_s}{l_c}} & if \;\; l_c <= l_s \end{cases} \tag{7.3}$$

A penalty named brevity penalty (BP) will be added if the translation result is too short. The brevity penalty (BP) is defined as Equation 7.3. l_c is the length of the machine translation result. l_s represents the length of the reference sentence. When there is more than one reference sentence, the length closer to the length of the machine translation result will be used as l_s. If the length of the machine translation result is less than the reference sentence, then the brevity penalty (BP) will be applied.

$$BLEU = BP \times exp(\sum_{n=1}^{N} logP_n) \tag{7.4}$$

Finally, we can calculate the BLEU score as Equation 7.4, where n is the number of grams. Here we consider the sum of the following four cases: n equal to 1, 2, 3 and 4, respectively, The final BLEU score will be calculated by an averaged P_n of the four cases.

7.2.2 Human Evaluation

Machine translation has a history of using human evaluation that predates automatic evaluation. A typical method is to analyze specific translation errors. The work of Vilar et al. (2006), Temnikova (2010) and Escribe (2019) analyzed several error categories including “missing word”, “word order”, “incorrect word”, “unknown words”, “punctuation”, “upper/lower case”. Popovic (2011) even proposed a toolkit to detect such error categories.

However, the type of translation error may depend on the specific language and content type. Such error analysis can only be utilized in the evaluation of a limited

number of translation directions. For instance, there is no need to utilize the error "upper/lower case" when evaluating Japanese-to-Chinese translation.

Aiming to utilize a general evaluation metric, people have tried to convert their evaluation judgments into numeric scores that can thus make the human evaluations comparable. This is not to say that human evaluations assigned numerical scores are not subjective, and can differ between one rater and another. Nevertheless, compared with subjective descriptions, it enables people to more intuitively judge the quality of translation using human evaluation.

After referring to the work of Denkowski & Lavie (2010), we adopt two human evaluation standards of the numerical type: Adequacy and Fluency.

1. Adequacy

   Adequacy measures how much of the content contained in the input sentence is conveyed by the translated sentence. Measuring this metric hence requires the evaluator to compare a translated sentence with its corresponding reference translation. The idea of this evaluation standard is similar to the concept of recall from information retrieval; that is, the evaluator needs to pay attention to judge the degree of translation preserved in the meaning of the input sentence.

2. Fluency

   While rating the fluency, the evaluator only needs to read one sentence (the translation result). In contrast, rating Adequacy requires the evaluator to compare two sentences. The scoring criteria for fluency encompass observing the translation result from such aspects: grammar, spelling, sentence style, and word order. These elements determine whether the translation result is understandable, or native-feeling, within its domain and language rules.

The scaled score of Adequacy and Fluency is an integer from 0 to 5. We define scores on this scale with the specific descriptions shown in Figures 7.1 and 7.2.

Table 7.1: Description of Adequacy

The degree of meaning preserved in translation

Score	Description
5	All meaning is preserved
4	Most meaning is preserved
3	Meaning of the main parts is preserved
2	Some meaning is preserved, but some meaning is lost.
1	Little meaning is preserved, most meaning is lost
0	None is preserved

Table 7.2: Description of Fluency

The degree of the understandability of the translation result

Score	Description
5	Easy to understand. sentence style is native
4	Easy to understand.
3	Enderstandable.
2	Enderstandable, but needs some conjectures.
1	Hard to understand.
0	Incomprehensible.

Human evaluation can be contrasted with the automatic evaluation metric such as BLEU. As explained in the previous subsection, BLEU is essentially a measure which looks for overlaps in n-grams appearance between the translation result and reference sentence. One situation to explain the disadvantage in this context is that since words with similar meaning, like “have” and “possess”, are different in appearance (are composed of different characters), BLEU will treat them as completely different. This means that similarities between words cannot be reflected correctly in the BLEU score.

Therefore, we utilized human evaluation as a supplement for automatic evaluation.

7.3 Results and Analysis

In this section, we present the quantitative results of the automatic evaluation and discuss the qualitative analysis of the results of human evaluation.

7.3.1 Analysis of Automatic Evaluation

This section mainly discusses and analyzes the result by utilizing the BLEU scores of automatic evaluation. The BLEU score, as defined in Equation 7.4, ranges from 0 to 1. The BLEU scores presented in this section, however, are the results of multiplying the original BLEU score by 100. The quality of the translation is considered proportional to the BLEU score. For the same test set, the larger BLEU score represents a better translation performance on it. In the tables of this section, the translation directions (Japanese-to-English and Japanese-to-Chinese) are referred to as ja-en and ja-zh. Each BLEU score in the tables of this section represents the result of each evaluation task.

We will analyze the automatic evaluation results based on the five perspectives as the following: (a) the performance of basic translation type (multi_5) of proposed system, (b) the effect of domain adaptation approaches, (c) the effect of model selection, (d) the effect of the model ensemble, and (e) the comparison of the two methods of model ensemble.

(a) Performance of the Basic Translation Type of the Proposed System

We analyze the evaluation results by referring to Tables 7.3 and 7.4. The rows of these tables refer to the translation type, while the columns designate the test set used.

Tables 7.3 and 7.4 show the evaluation results (BLEU scores) of the basic translation type (multi_5) of the proposed multi-domain translation system. The translation type named Google refers to the translation result obtained using the Google Translate service.[5]

Through Table 7.3, we can see that for the ja-en translation direction, the results of multi_5 on the News, Travel, IT and Mixed test sets are better than the results of baseline, and also better than the results of Google. For instance, on the News test set, multi_5 was about 6 BLEU points higher than the baseline. For the ja-zh

[5]It must be mentioned that commercial machine translation services are updated continuously. We utilized Google's Google Translate service to translate the test set during October, 2019.

Table 7.3: Evaluation: Basic performance of the system (ja-en)

	Google	baseline	multi_5
General	37.52	37.66	36.89
News	40.77	37.10	43.09
Travel	30.14	26.63	30.57
IT	33.61	22.46	35.92
Mixed	36.90	32.75	37.67

Table 7.4: Evaluation: Basic performance of the system (ja-zh)

	Google	baseline	multi_5
General	34.45	32.65	32.76
News	39.22	36.83	40.34
Travel	27.90	25.46	29.05
IT	30.74	20.08	35.17
Mixed	34.07	30.68	36.88

translation direction, a similar result was observed in Table 7.4. Multi_5 is the basic translation type of the proposed system. It meets the premise of our proposed multi-domain translation that translates without using the domain information of the input text. This result illustrates the effectiveness of our system in translating domain text for ja-en.

On the other hand, for both translation directions, the result of multi_5 on the General Test was close to the baseline. This result showed that even while our proposed system reduces the translation quality slightly outside the specific domains, it still also performs well.

We also recognized that the result of baseline on the IT test set is the lowest result among all results of the baseline. It shows that IT could be the most challenging domain for baseline. On the other hand, from baseline to multi_5, the results on the IT test set achieved the greatest improvement among all test sets. The result improved about 13 points of BLEU score in 7.3 and about 15 points of BLEU score in 7.4, respectively. This comparison demonstrates that among the three domain-specific test sets, IT is the most suitable for applying multi-domain translation.

Table 7.5: Evaluation: Domain adaptation approaches (ja-en)

	baseline	ALL	Fine-Tuning	Mixed Fine-Tuning	stacking	data selection
News	37.1	41.57	39.34	40.09	37.87	39.24
Travel	26.63	29.26	31.01	32.67	29.30	30.76
IT	22.46	33.89	29.94	35.07	31.68	35.39

Table 7.6: Evaluation: Domain adaptation approaches (ja-zh)

	baseline	ALL	fine-tuning	mixed fine-tuning	stacking	data selection
News	36.83	39.45	39.14	39.24	38.87	37.90
Travel	25.46	26.93	27.22	28.08	26.48	27.06
IT	20.08	28.38	32.02	34.67	31.87	32.16

As a simple conclusion, for both translation directions, since multi_5 has the highest score on the News, Travel, IT, and Mixed test set, the effectiveness of the proposed multi-domain translation system had been validated.

(b) Effect of Domain Adaptation

Since we utilized five domain adaptation methods to train the domain-specific model in Chapter 3, we need to analyze whether the domain-specific model is more effective than the baseline on the specific domain. The rows of Tables 7.5 and 7.6 refer to the domain-specific models or the baseline model. The columns of Tables 7.5 and 7.6 refer to the domain-specific test sets.

We compare the performance of five specific-domain models of each domain horizontally. For both translation directions, all the specific-domain models performed better than the baseline. For instance, for translation direction ja-en, on the Travel test set, the best result was mixed fine-tuning, with a BLEU score of 32.67, and the lowest was ALL, with 29.26. The lowest one was still about 3 points of BLEU score higher than the baseline. This proved the effectiveness of using domain adaptation to improve the translation quality on the specific domain. Especially for the approach of data selection proposed in this thesis, it proved to be higher than some other domain adaptation approach, except on the News test

set in Table 7.6. For instance, on the IT test in Table 7.5, data selection is the best result.

However, since we noticed that on some domains, the results of the five domain-specific models vary greatly, it is difficult for us to assess which domain adaptation method was generally best on all three domains test sets. For instance, in Table 7.5, on the News test set, the result of stacking was 37.87; the result of ALL was 41.57. The difference is close to 4 BLEU points. On the other hand, in Table 7.5, on the same News test set, the difference between the five domain-specific models is roughly 1 BLEU point. This problem motivates us to propose multi-domain translation which dynamically selects several suitable models from multiple models and uses them to translate through model ensemble.

As a simple conclusion, the domain-specific model trained by domain adaptation had better results than the baseline model. However, we have not found the most effective method from these five types of methods.

(c) Effect of Model Selection

In this thesis, we adopted the approach of model selection to select suitable models from multiple models and used the model ensemble approach for translation. Here, we recorded the highest BLEU score among five domain specific models of Tables 7.5 and 7.6 as the no_selection in Tables 7.7 and 7.8. We need to analyze the effectiveness of model selection by comparing the translation performance between no_selection and the translation type domain_1/multi_1.

For the specific domains (News, Travel, IT), since we know the domain information, we can, in theory, directly use the domain_1 translation type to select the best model. On the other hand, using multi_1 is akin to simulating choosing the best model without knowing the domain information.

The rows of Tables 7.7 and 7.8 refer to translation types. The columns of Tables 7.7 and 7.8 refer to the Test sets. Since there is no domain-specific model corresponding to the General/Mixed test set, we record the empty result using the mark “x”.

Table 7.7: Evaluation: Model selection (ja-en)

	baseline	no_selection	domain_1	multi_1	multi_5
General	37.66	×	×	37.50	36.89
News	37.1	41.57	42.24	41.79	43.09
Travel	26.63	32.67	32.03	29.07	30.57
IT	22.46	35.39	35.78	32.72	35.92
Mixed	32.75	×	×	34.47	37.67

Table 7.8: Evaluation: Model selection (ja-zh)

	baseline	no_selection	domain_1	multi_1	multi_5
General	32.65	×	×	31.91	32.76
News	36.83	39.45	39.62	37.95	40.34
Travel	25.46	28.08	28.11	26.33	29.05
IT	20.08	34.67	34.64	33.57	35.17
Mixed	30.68	×	×	32.13	36.88

By comparing the results between no_selection and domain_1 horizontally, we found that the two results are close on each domain-specific test set. For instance, in Table 7.7, on the IT test set, the result of no_selection is 35.39, and the result of domain_1 is 35.78. It proved the effectiveness of the model selection approach of domain_1.

On the other hand, the result of multi_1 is usually lower than the result of no_selection. Since there were 15 models available for the translation type of multi_1, the range of candidate models was more than the translation type of domain_1. It was hard to select the appropriate model for each sentence accurately. For the same test and translation direction, in Table 7.7, the result of multi_1 is 3 points BLEU points lower than the result of no_selection. From this phenomenon, we can see the shortcoming of the model selection approach of multi_1 on the News, Travel, and IT test set.

We further compared domain_1/multi_1 with multi_5. Except the General test set in Table 7.7, for both translation directions, we found an improvement from multi_1 to multi_5 on each Test set. For instance, in Table 7.8, multi_5 improves 3 BLEU points on the Travel test set. In this example, multi_5 is also higher

Table 7.9: Evaluation: Domain ensemble (ja-en)

	baseline	domain_1	domain_3	domain_5	multi_1	multi_5
News	37.1	42.24	42.88	42.87	41.79	43.09
Travel	26.63	32.03	33.45	32.78	29.07	30.57
IT	22.46	35.78	38.10	36.32	32.72	35.92

Table 7.10: Evaluation: Domain ensemble (ja-zh)

	baseline	domain_1	domain_3	domain_5	multi_1	multi_5
News	36.83	39.62	39.54	39.77	37.95	40.34
Travel	25.46	28.11	32.42	31.80	26.33	29.05
IT	20.08	34.64	36.99	36.68	33.57	35.17

than no_selection/domain_1. This suggests that selecting multiple models could improve the effectiveness of model selection.

As a simple conclusion, the approach of model selection presented here is effective.

(d) Effect of Model Ensemble

In this thesis, we adopted the approach of model selection to select suitable models from multiple models and used the model ensemble approach for translation. We need to analyze the effectiveness of the model ensemble.

The rows of Tables 7.9 and 7.10 refer to translation types. The columns of Tables 7.9 and 7.10 refer to the Test sets.

Ensembling multiple models improved the result of using one model. For instance, in Table 7.10, on the IT test set, the BLEU score of domain_1 was 35.78, and the BLEU score of domain_5 was slightly higher at 36.32. Similarly, in the same line, the BLEU score of multi_1 was 32.72, and the BLEU score of multi_5 was 35.92.

The comparison of the results of multi_1 and multi_5 might suggest that if we could collect as many models as possible for the model ensemble, we would get better results. However, this is not the truth. We can find that sometimes the result of domain_3 was better than domain_5. For example, for ja-en, on

Table 7.11: Evaluation: Weighting strategies (ja-en)

	domain_5 (uniform)	domain_5 (weighted)	multi_5 (uniform)	multi_5 (weighted)
News	41.64	42.87	40.11	43.09
Travel	30.34	32.78	28.25	30.57
IT	34.92	36.32	32.06	35.92
Mixed	×	×	32.44	37.67

Table 7.12: Evaluation: Weighting strategies (ja-zh)

	domain_5 (uniform)	domain_5 (weighted)	multi_5 (uniform)	multi_5 (weighted)
News	38.42	39.77	37.44	40.34
Travel	29.54	31.80	27.25	29.05
IT	33.91	36.68	34.52	35.17
Mixed	×	×	34.98	36.88

the IT test set, the BLEU score of domain_3 was 33.45, and the BLEU score of domain_5 was slightly lower, which was 36.32. Therefore, when translating with the type domain_n, it is necessary to avoid ensembling all domain-specific models. Comparing the results of domain_3 and domain_5, shown in Tables 7.9 and 7.10, we can see that using three models for model ensemble may perform better than using five models.

(e) Comparison of the Two Weight Strategies for Model Ensemble

When ensembling multiple domain models, we discussed two weighting strategies in Section 4.3.2 . From Tables 7.11 and 7.12, the strategy of weighted is better than uniform. The results of weighted were generally higher than the results of uniform for about 1 or 2 BLEU points.

Therefore, we set the method of weighted as the default weighting strategy. It is important to note that unless otherwise stated, the translation types using model ensemble are all based on the default weighting strategy.

7.3.2 Analysis of Human Evaluation

As already described, (a) Adequacy and (b) Fluency are the two aspects of human evaluation. These results are discussed in turn.

Table 7.13: Evaluation: Adequacy (ja-en)

	baseline	domain_1	domain_3	domain_5	multi_1	multi_5
General	4.62	X	X	X	4.48	4.50
News	4.22	4.69	4.71	4.85	4.69	4.88
Travel	3.96	4.43	4.76	4.53	4.38	4.64
IT	3.02	4.20	4.77	4.63	4.09	4.54
Mixed	3.73	X	X	X	4.38	4.68

Note: The highest result on each test set is underlined.

Table 7.14: Evaluation: Adequacy (ja-zh)

	baseline	domain_1	domain_3	domain_5	multi_1	multi_5
General	4.60	X	X	X	4.45	4.50
News	4.06	4.32	4.51	4.38	4.28	4.36
Travel	3.88	4.07	4.42	4.44	4.02	4.16
IT	3.06	3.99	4.20	4.19	3.92	4.15
Mixed	3.66	X	X	X	4.07	4.22

Note: The highest result on each test set is underlined.

(a) Adequacy

As mentioned in Section 7.2.2, Adequacy measures how much of the information of the input sentence is retained after the translation process. The Adequacy values in Tables 7.13 and 7.13, which are averaged scores from two human evaluators, are scaled from 0 to 5.[6] The high Adequacy value of a translation type means that the translation result of the translation type could have fewer problems. When the evaluator found a translation error such as a missing word or incorrect word translation, they will decrease the Adequacy value.

For both translation directions on all Test sets (except General), the basic translation type of our proposed system multi_5 is higher than the baseline. On some specific Test sets, multi_5 could be the best result (scores with underline) among all the translation types. For example, in Table 7.13, multi_5 is the best translation type on the News test set. It proved the effectiveness of the basic translation type of the proposed system.

[6]For the General and Mixed test set, the result of domain_n is empty.

We also analyze the translation type order after sorting the values of Adequacy from large to small. For example, for the ja-en translation direction (Table 7.13), on the IT test set, the descending order of translation types with respect to the Adequacy score could be:

$$\text{domain\_3} > \text{domain\_5} > \text{multi\_5} > \text{domain\_1} > \text{multi\_1} > \text{baseline}$$

This reflects the effectiveness of the model selection since multi_1/domain_1 is better than the baseline, as well as the effectiveness of the model ensemble since domain_5/multi_5 is better than domain_1/multi_1. Furthermore, as domain_3 was the best translation type on almost test set cases, we suggest ensembling three models instead of five models, which is a similar conclusion to that arrived at in the results of automatic evaluation.

(b) Fluency

As mentioned in Section 7.2.2, Fluency measures how difficult it is for humans to understand the translation result. The Fluency figures in Tables 7.15 and 7.16, which are averaged scores from two human evaluators, are scaled from 0 to 5.[7] When the evaluator finds a translation error like wrong word order, they will decrease the Fluency score.

For both translation directions on all Test sets (except General), the basic translation type of our proposed system, multi_5, is higher than the baseline. On some specific Test sets, multi_5 could achieve the highest result among all the translation types. For example, in Table 7.15, multi_5 is the best translation type on the IT test set. These results proved the effectiveness of the basic translation type of the proposed system.

We also analyze the translation type order after sorting the scores of Fluency from large to small. For example, for the ja-en translation direction (Table 7.16),

[7] As before, for the General and Mixed Test set, the result of domain_n is empty.

Table 7.15: Evaluation: Fluency (ja-en)

	baseline	domain_1	domain_3	domain_5	multi_1	multi_5
General	3.87	X	X	X	3.83	3.80
News	3.46	3.93	4.23	4.08	3.98	4.16
Travel	3.08	3.56	3.60	3.61	3.47	3.82
IT	2.77	3.38	3.42	3.46	3.37	3.56
Mixed	3.10	X	X	X	3.60	3.84

Note: The highest result on each test set is underlined.

Table 7.16: Evaluation: Fluency (ja-zh)

	baseline	domain_1	domain_3	domain_5	multi_1	multi_5
General	3.53	X	X	X	3.57	3.59
News	3.32	3.88	3.96	4.04	3.91	4.11
Travel	3.01	3.64	3.91	3.82	3.87	3.90
IT	2.84	3.23	3.49	3.35	3.20	3.44
Mixed	3.05	X	X	X	3.68	3.81

Note: The highest result on each test set is underlined.

on the News test set, the descending order of translation type with respect to the Fluency score could be:

multi_5 > domain_5 > domain_3 > multi_1 > domain_1 > baseline

The ordering reflects the effectiveness of model selection. This is because multi_1/domain_1 is better than the baseline. The effectiveness of the model ensemble is also confirmed, since domain_5/multi_5 is better than domain_1/multi_1. However, there are three cases where domain_3 is better than domain_5, and three cases where domain_5 is better than domain_3. Contrasting to the conclusion of the automatic evaluation, these results suggest not ensembling three models instead of five.

For 5 test sets and two translation directions, there are 10 cases. We found that in these 10 cases, the translation type multi_5 achieved the highest results six times. In other cases, it was only slightly lower than the optimal result. This suggests that if we need to emphasize fluency, multi_5 may be the best translation

type.

7.4 Case Studies

We utilize some specific examples to examine the overall effectiveness of our proposed system.

We also examine the translation results obtained from different translation types to investigate whether the proposed approaches of domain adaptation, model selection and domain ensemble, can improve the translation performance.

Hence, we compared the translation results from two aspects: comparing translation results from different domain adaptation methods and comparing translation results from different translation types.

Through the comparison of actual example sentences in this case study, we found that one of the most significant improvements from baseline to multi_5 is that the translation choice for specific words has been improved. Especially some words that were treated as unknown words in the baseline have been correctly translated.

7.4.1 Comparing Five Adaptation Approaches

In order to examine the effectiveness of the domain adaptation approaches utilized in this thesis, in this section, we compared and analyzed the results translated by the baseline and the domain-specific models (ALL, fine-tuning, mixed fine-tuning, stacking, data selection) with some specific examples.

(12) a. ログオン認証パケットの不達や遅延によるアクセス不能問題を起こしやすい。

b. (baseline) It is easy to cause inaccessibility problem due to missed or delayed logon認証パケット

c. (ALL) It is easy to cause an inaccessible problem due to irregularities and delays in the log-on authentication packet.

d. (fine-tuning) It will easily cause inaccessibility problems due to missed or delayed ログオン logon authentication packets.

e. (mixed fine-tuning) It is easy to cause inaccessibility problem due to non-delivery or delay of logon authentication packet.

f. (stacking) Prone to inaccessible problems due to unreachability or delay of logon authentication packets.

g. (data selection) It is likely to cause an inaccessible problem due to the lack or delay of logon authentication packets.

In Example (12), where the input sentence (sentence-a) is from the IT test set, the general meaning of the Japanese sentence (sentence-a) is that "It is easy to cause access failures due to non-delivery or delay of logon authentication packets.". The translation type baseline could not translate "認証パケット" (authentication packet) and left it as appeared in the input. For the other translations utilizing domain adaptation, they almost correctly translated the word "認証パケット" into "authentication packets". The basic translation type of our proposed system is to select models from domain-specific models and ensemble them. Since all the domain-specific model improved the translation quality, it laid a good foundation for our system. If the process of model selection and model ensemble is effective, the effectiveness of the entire system could be confirmed.

Furthermore, we could observe the difference between the five domain-specific models. In terms of translation word choice, "不達" was translated into "missed", "irregularities", "non-delivery", "unreachability", and "lack", respectively, most of them were adequate.

From the aspect of syntax, almost all the results chose the syntax "it is likely/easy", while the stacking (sentence-f) erroneously chose a more difficult word, "Prone". This mistake may be due to the relatively small number of corpus data utilized in each training epoch of the stacking method, which makes it easier for the model to learn some extreme parameters which are insufficient for general translation.

From the aspect of mistration, fine-tuning (sentence-d) even had a translation

error. It had already translated “ログオン” to logon. However, it still output an unknown word “ログオン”. Erroneous translation like this might be attributed to the insufficient functionality of the attention mechanism, but the further investigation should be needed.

From the following two practical examples, we can also assess that the model after domain adaptation was more adequately translated domain-specific terms.

(13) a. 同一のグループプロパティ名を持つマテリアル内の全パラメータは、インスタンスのカテゴリ配下にリストされます。

b. (baseline) 在 实例类别下列出了具有相同 グループプロパティ 名称的材 质中的 所有参数 。

c. (ALL) 具有相同组性 质名称的素材内的所有参数，在 实例的类别下被列表。

d. (fine-tuning) 在同一组属性名称的材料中的所有参数将被列入 实例类别下方。

e. (mixed fine-tuning) 在 实例类别下列出了拥有相同组属性名称的元素中的所有参数 。

f. (stacking) 材料中具有相同组属性名称的所有参数都列在 实例类别下。

g. (data selection) 具有相同组属性名称的材料中的所有参数都列在 实例的类别下。

Some technical jargons that were often treated as unknown words in the baseline were often better handled in the domain specific model. The differences between the various domain adaptation methods are manifested in various aspects: different syntactic structures, different translation choices, and whether there have been mistakes or omissions.

In Example (13), where the input sentence (sentence-a) is taken from the IT test set, the general meaning of the Japanese sentence (sentence-a) is that “All parameters in a material with the same group property name are listed under the instance category”. The translation type baseline did not translate the professional

term “グループプロパティ” (group property) well. For the other translations utilizing domain adaptation, they almost correctly translated the word “グループプロパティ” into “组属性” (group property) or a similar word “组性 质” (group property). Since the five domain-specific models improved the translation quality compared to the baseline, they laid a good foundation for our system to execute the translation type domain_n or multi_n.

Furthermore, we can analyze the difference between five domain-specific models. In terms of translation word choice, “マテリアル” (material) was translated into similar words “材 质” (material), “材料” (material), and “素材” (material), respectively.

In terms of syntax, the differences were whether the passive marker was used or not. Some domain-specific models utilized the Chinese passive marker “被”, such as ALL, and fine-tuning. Other domain-specific models did not use a passive symbol, which is more native-sounding in Chinese. Besides, ALL translated the verb “リストする” (list) into “列表” (list), but “列表” in Chinese is used more as a noun. The translation here was considered inappropriate.

As a simple conclusion, the domain-specific model ALL in this example is relatively weak. This situation is different from Example (12), where the weakest model is stacking. In other words, several specific domain methods do not have an absolute advantage or disadvantage. This showed the necessity that selecting suitable models based on each sentence.

7.4.2 Comparing Different Translation Types

In order to examine the effectiveness of the proposed approach of model selection and model ensemble in this section, we compared and analyzed the results of four translation types (baseline domain_1, multi_1, multi_5) with some specific examples.

(14) a. エアコン、専用バスルーム、液晶衛星テレビ、無料 Wi-Fiが備わる客室です。

b. (baseline) Air conditioners, dedicated bathrooms, LCD satellite TVs , free Wi-Fi are the 備わる guest rooms.

c. (domain_1) This guest room has air-conditioner, private bathroom, LCD satellite TV and free Wi-Fi.

d. (multi_1) This air-conditioned room has a private bathroom, LCD satellite TV and free Wi-Fi.

e. (multi_5) This room has air conditioner, private bathroom, LCD satellite TV and free Wi-Fi.

In Example 14, where the input sentence (sentence-a) is taken from the Travel test set, the general meaning of the Japanese sentence (sentence-a) is that "The guest room has an air-conditioner, private bathroom, LCD satellite TV and free Wi-Fi". The other sentences in Example 14 correspond to the English translation result using the translation type baseline, domain_1, multi_1, and multi_5.

Theese results suggest that the baseline (sentence-b) failed to correctly capture the syntactic structure of the input in Japanese. The Japanese sentence (sentence-a) is a noun phrase. However, the baseline (sentence-b) translated it as a pattern "A が B -> A is B". For other translation types that utilized the approaches proposed in this thesis, no such mistake was observed. sentence-a is correctly translated into the pattern "A has B". This demonstrated the effectiveness of our proposed system.

We also noticed that multi_1 (sentence-d) made a mistake utilizing "エアコン" (air-conditioner) as the modifier of "客室"（guest room). It made the translation error "air-conditioned room". Such a mistake was caused by model selection. Since the words "液晶" (LCD) and "Wi-Fi" often appear in the corpus of the IT domain, when multi_1 selected the model, it wrongly selected an IT domain-specific model. On the other hand, multi_5 (sentence-e) correctly translated sentence-a. It showed the necessity of using the model ensemble since the model selection of multi_1 is not the best choice in some cases.

(15) a. 問題:複製されたマーク値を削除しますか?

b. (baseline) Problem :複製 Delete Mark Value?

c. (domain_1) Problem: do you want to delete a duplicate mark value?

d.（multi_1）Issue : How to delete duplicated mark values?

e.（multi_5）Problem: delete duplicated mark values?

In Example 14, where sentence-a is taken from the IT test set, the general meaning of the Japanese sentence (sentence-a) is "Problem: Do you delete duplicated mark values?". By comparing the baseline (sentence-a) and other translation types (sentence-b to sentence-e), it is clear that other translation types solved the mistranslation of the word "複製" (duplicate) caused by the baseline (sentence-b). This example exemplified the effectiveness of our proposed system.

Furthermore, by comparing translation types of domain_1 and multi_1, we found that an inappropriate translation "how to" appears in the result of multi_1. Such mistakes would be caused by model selection. Hence we need to utilize more than one model by using a model ensemble to improve the quality when the translation type multi_n is used. This solution was validated by the case where multi_5, the basic translation type of our proposed system, has fixed this error.

(16) a. ギリシャなど欧州に端を発した財政危機の波紋が広がり、我が政府や政界も財政健全性管理に向けた対策作りに苦心している。

b.（baseline）希 腊和欧洲的财政危机正在蔓延，我国政府和政治界都在努力制定财政 稳健措施。

c.（domain_1）由于起源于希 腊的欧洲金融危机的 连锁反 应，政府和政治都在努力制定措施来管理财政 稳健性。

d.（multi_1）由希 腊等欧洲引 发的财政危机的影 响正在蔓延，我国政府和政界也在努力制定针对财政健全性管理的 对策。

e.（multi_5）随着希 腊等欧洲国家引 发的财政危机的影 响的 扩散，我国政府和政治圈也在为制定财政健全性管理而绞尽脑汁。

In Example 16 sentence-a is taken from the News test set, the general meaning of the Japanese sentence is "With the ripple effects of the financial crisis originating from Europe, such as that of Greece, our government and politics are struggling

to make a strategy for fiscal soundness management.". The other sentences in Example 14 correspond to the Chinese translation results using the translation types baseline, domain_1, multi_1, and multi_5.

We first focus on the translation choice of the underlined Japanese phrase "端を発した" (originating from). Translation type baseline (sentence-b) missed the phrase "端を発した". On the other hand, translation type multi_5 (sentence-e) accurately translated "端を発した" to "引 发" (caused by). It showed the effectiveness of the basic translation type of our proposed system. What is more, domain_1 (sentence-c) and multi_1 (sentence-d) also accurately translated "端を発した" to "起源于" (originating from) and "引 发" (caused by), respectively. This proves that the domain adaptation improved the performance, and that model selection can select a suitable model for translating sentence-a.

We also focus on the translation choice of the underlined Japanese phrase "対策作りに苦心している" (struggling to make a strategy). Except for multi_5, other translation types all translated "対策作りに苦心している" to "努力制定" (struggle to plan something). The object of the verb "努力制定" (struggle to plan something) came after the verb, a typical SVO[8] order. On the other hand, multi_5 utilized an advanced Chinese expression "XXX绞尽脑汁" (struggle to do something) to translate "XXX に苦心している". This advanced use of vocabulary made the translation more native-like. The object of the verb "XXX绞尽脑汁" (struggle to do something) came before the verb, a typical SOV[9] order similar to Japanese. Hence, we realized that comparing with multi_5 and multi_1/domain_1, utilizing five models with a model ensemble improved the translation quality.

(17) a. 協会は、新たに八百長容疑が明らかになった 14 人の力士に対しても、独自調査を進めている。

b. (baseline)该协会正在 对800 人的罪名的 14 名相 扑选手 进行独立调查。

c.（domain_1）协会 还对被新 查出造假嫌疑的 14 名力士 进行了独立

[8]SVO: denotes the word order of a language, which is Subject-Verb-Object in this case.
[9]SOV: the word order is Subject-Object-Verb.

调查。

d. (multi_1)该协会 还对14 名最近被 确定为有嫌疑的 摔跤手 进行了调查。

e. (multi_5)协会 对14 名新 查明的 涉嫌假比 赛的相 扑力士 进行了独立调查。

In Example 17 sentence-a from the News test set, the general meaning of the Japanese sentence is "The association is also conducting an independent investigation on 14 sumo wrestlers who have been newly identified as throwing the match/doing fixed games".

We focus on the translation choice of the underlined Japanese word "八百長" (throw a match/fixed game). The baseline translated "八百長" as a number word phrase "800 人" (eight hundred). This shows that the baseline model did not recognize that the word "八百長" was a special word. The translation type multi_1 (sentence-b) translated phrase "八百長容疑" (suspicion of throwing the match) to "造假嫌疑" (suspicion of fabricating), which was relatively close to the true meaning, but still subtly different. The translation of multi_1 (sentence-c) missed the "八百長". This may indicate that translation type multi_1 (sentence-c) could not select a suitable model for translating "八百長". On the other hand, domain_1 selected a better model than multi_1.

The advantage of a model ensemble is that it can utilize multiple models comprehensively. The translation type multi_5 (sentence-e) accurately translated "八百長" into "假比 赛" (fixed game). It seems that multi_5, ensembling five models, could overcome the lack of the one model selected by multi_1/domain_1.

7.5 Chapter Summary

In this chapter, we used automatic and human evaluation methods to evaluate the performance of the multi-domain translation system proposed in this thesis on multiple test sets from five perspectives. Since the results of the basic translation type multi_5 of the proposed system was higher than the baseline and close to

domain_3, the effectiveness of our multi-domain translation was confirmed.

From these evaluation results, we also obtained specific insights and uncovered issues that might be addressed by future work.

7.5.1 Insights

1. The necessity of using domain specific models:
   We recognize that the domain specific model performs better than the baseline model in translating specific domain text. What is more, data selection, the domain adaptation approach proposed by this thesis, has been proven to be effective, since it could get a better result than other domain adaptation approaches such as ALL and stacking.

2. The necessity of model selection and model ensemble:
   For different domain-specific test sets, the domain-specific model which provides the best result varies. Thus we must note that we could not confirm the best domain adaptation approach for each domain at this stage of research.

   We selected suitable models and ensembled them. Since the evaluation showed that domain_1/multi_1 is better than only using the single model, which is the best model among five domain-specific models, the effectiveness of model selection has been demonstrated. Furthermore, by comparing with domain_1/multi_1 and domain_3/multi_5, we proved that model ensemble might improve the translation quality.

3. Recommended translation type:
   When translating input text without its domain information, we select models from all the candidate models. In this case, the evaluation results show that the translation type multi_5 can provide better results than other translation types. If the domain information is given, we could reduce the range of candidate models from all models to models of a specific domain. In this case, the evaluation results show that the translation type domain_3 can provide better results than other translation types. This suggests that there

could be an optimum number of models to be adopted, but it is difficult to decide the number in advance.

4. Specific improvement to approaches proposed in this thesis:
   The domain-specific model trained by the domain adaptation approach can significantly improve the performance of the model in translating professional vocabulary. Furthermore, on the view of translation choice among similar words, it showed that after model ensemble, the system might choose the more advanced or native word.

7.5.2 Uncovered Issues

As an ideal situation, we envision that the multi_n type translation approach could completely replace the domain_n type translation approach. To this purpose, we have discovered and summarized some issues that might initiate future work. These issues can be discussed in the following two perspectives.

1. multi_1 vs. domain_1

   In theory, for a specific domain, if we select only one suitable model, the result of domain_1 should be the same as multi_1. However, as was clearly shown by the results of the automatic evaluation, the results of domain_1 and mutli_1 are not the same. This implies that mutli_1 had selected a model outside of the target domain. Furthermore, since multi_1 scored slightly lower than domain_1, it showed that there is space for improving the multi_n translation type if we can improve the precision of model selection. Adopting more effective criteria of model selection could be a solution.

2. Addition of new domains

   In the framework of our proposed system, it is easy to add new domain-specific models without re-training all existing models. On the other hand,

adding new domains will increase the workload of the model selection and model ensemble. As a result, it could influence the translation speed of the translation type multi_n, especially for translating large amounts of text.

The results here showed that our basic translation type multi_5 is close to but not significantly higher than domain_3 (the best translation type of domain_n) through the evaluation. Therefore, we must measure and balance the cost/performance of multi_5 between translation quality and translation speed while adding new domains. This entails speeding up the translation speed to keep up with the increased number of models.

Chapter 8

Conclusion

This chapter will briefly review and summarize the research contributions of this thesis (Section 8.1) and then look forward to some related topics that are feasible avenues towards further improvements (Section 8.2).

8.1 Summary

Neural Machine Translation (NMT) is an effective machine translation technique developed in recent years. Its translation quality has far surpassed traditional machine translation, such as statistical machine translation (SMT) or rule-based machine translation (RBMT). Nevertheless, NMT is not perfect in all aspects. For a specific domain, if the NMT model is not trained with enough corpus data of the domain, it could reduce the translation quality when translating text from the domain. Therefore, it is often necessary for NMT systems to prepare different domain-specific models for texts of different domains.

However, in the actual scenario of translation applications, there are cases where the text domain cannot be quickly determined. It is hard to detect which translation model to use for translation, and thus impossible to avoid the risk of a domain detection error. For such a case, we switched to apply the translation method of multi-domain translation.

Multi-domain translation is a method for utilizing multiple domain-specific translation models, and its role is to determine the most suitable translation model

for translating the input sentence correctly.

Therefore, this thesis proposed a multi-domain translation method, constructed an effective neural machine translation system with the proposed method, and verified the effectiveness of the system.

The contributions of this thesis can be summarized as follows.

1. Proposed a framework and system for multi-domain translation
   The proposed system is composed of many domain-specific models. During the translation process, the system will find and select the appropriate domain-specific model for the input sentence, and then utilize the selected models by model ensemble to obtain a high-quality translation.

   From the perspective of the system architecture, the multi-domain translation system framework proposed in this thesis also has other advantages, that is, it does not require an additional domain detection module to determine the domain of the input sentence.

2. Proposed approaches to prepare candidate domain-specific translation models
   In this contribution, we operated under the assumption that the domain-specific translation model in the proposed system is specialized for the specific domain. The specialization for the domain relies on utilizing the approach of domain adaptation.

   This thesis applied five kinds of domain adaptation approaches to construct a domain-specific model. Among these approaches, in addition to drawing on some traditional domain adaptation approaches, this thesis also proposed an approach of data selection based on the semantic similarity of aligning score. The models trained by the data selection approach have been proven to perform well in the translation evaluations conducted. Compared to models that have not utilized any approaches of domain adaption, domain-specific

models using data selection improved the translation quality by 2-4 BLEU points.

3. Proposed an approach for model selection based on perplexity
   Perplexity is a measure to evaluate the difficulty of the model producing the correct translation. Given an input sentence, a high perplexity of the model means a high difficulty of translation. Thus, perplexity can be used to determine that the model is not an appropriate for the input sentence. Conversely, when the perplexity of the model is low, it is considered appropriate. In this way, even if the domain of the input sentence was unknown, the most appropriate model could be selected for the input sentence by measuring the translation difficulty of the model.

   The evaluation shows that using the translation type multi_1/domain_1 could obtain a higher evaluation result than using only a single model, which is the best domain-specific model on the present domain. This indicates that using the perplexity-based approach can select suitable model for each sentence.

4. Verified the effect of the model ensemble
   The model ensemble uses each model comprehensively and produces better translations than a single model. We were also weighting each model based on perplexity to influence adjusting the effect of the model ensemble. Through the experimental results, we found that the translation obtained by the model ensemble performed better than the translation obtained from a single model.

   More specifically, when facing input text lacking domain information, the best choice is to ensemble the five best models, as multi_5 is the basic translation type of our proposed system. On the other hand, if we can know the domain information, an alternative solution is the translation type domain_3.

As a conclusion, we have finally confirmed the effectiveness of the framework

of the multi-domain translation system proposed in this thesis. It can effectively solve the problem where even if it is not possible to determine the domain of the input text quickly, it can give a suitable model for the input text and finally generate a high-quality translation result.

8.2 Future Work

At the same time, we can also look forward to two new topics for future study. Research on these two topics successfully is expected to further improve performance of a multi-domain translation system.

1. High speed translation

   In the system framework proposed in this thesis, when we select models, we need to use each candidate model to translate, calculate the perplexity value, and then ensemble the selected model. The system repeats this process for each sentence. If we can reduce the number of iterations needed to execute this process, it could speed up the translation. This kind of strategy makes sense when translating a large volume of text.

   From the perspective of executing the translation program, it takes time to load the model. The suitable translation models required for each sentence are not the same. Therefore, the critical point of a programming implementation is that there is no need to reload the model between the translation of the previous sentence and the following sentence.

   From the perspective of improving the process, we could utilize domain detection. Although we cannot utilize domain detection directly to select models from the perspective of ensuring translation accuracy, we can, however, use a mechanism similar to domain detection to pre-categorize the input sentences. If it can reduce the number of required models among the sentences of the same category, the execution times of subsequent processes could be significantly simplified.

2. Improving the accuracy of model selection

   In future research, improving the accuracy of model selection is an important direction. Therefore, another new topic in future research is using some other measure for selecting the translation models instead of perplexity.

   In the approaches proposed in this thesis, we used the perplexity measure as the criterion to select the appropriate model. In contrast, one possible idea is using a direct indicator such as the quality of the translation result.

   However, in order to measure or to evaluate the quality of the translation result, we often need a reference sentence which may not exist in the real case. A feasible and alternative method is to find sentences that are similar to the input sentence from a bilingual corpus. Then use their corresponding translations to generate substitutes for the reference sentences.

   To verify whether this method is feasible and effective, the following four sub-topics should be pursued.

   (a) How to identify the sentences similar to the input sentence.

   (b) How to get the substitute for the reference sentence.

   (c) How to measure translation quality using the substitute for the reference sentence.

   (d) How to choose a translation model based on the measure of translation quality.

Bibliography

Al-onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F.-j., Purdy, D., Smith, N., & Yarowsky, D. (1999). Statistical machine translation. *Final Report JHU Workshop 1999*.

ALPAC (1966). *Language and Machines: Computers in Translation and Linguistics*. Washington, DC: The National Academies Press.

Axelrod, A., He, X., & Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (pp. 355–362). Edinburgh, Scotland, UK.: Association for Computational Linguistics.

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the 2015 The International Conference on Learning Representations (ICLR)*.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., & Tamchyna, A. (2014). Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation* (pp. 12–58). Baltimore, Maryland, USA: Association for Computational Linguistics.

Bojar, O. & Tamchyna, A. (2011). Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation* (pp. 330–336). Edinburgh, Scotland: Association for Computational Linguistics.

Brill, E. & Moore, R. C. (2000). An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics* (pp. 286–293). Hong Kong: Association for Computational Linguistics.

Britz, D., Le, Q., & Pryzant, R. (2017). Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation* (pp. 118–126). Copenhagen, Denmark: Association for Computational Linguistics.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2), 263–311.

Chen, S. F. & Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *34th Annual Meeting of the Association for Computational Linguistics* (pp. 310–318). Santa Cruz, California, USA: Association for Computational Linguistics.

Cheng, Y., Xu, W., He, Z., He, W., Wu, H., Sun, M., & Liu, Y. (2016). Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1965–1974). Berlin, Germany: Association for Computational Linguistics.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)* (pp. 263–270). Ann Arbor, Michigan: Association for Computational Linguistics.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, 33(2), 201–228.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). Doha, Qatar: Association for Computational Linguistics.

Chu, C., Dabre, R., & Kurohashi, S. (2017). An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 385–391). Vancouver, Canada: Association for Computational Linguistics.

Chu, C., Dabre, R., & Kurohashi, S. (2018). A comprehensive empirical comparison of domain adaptation methods for neural machine translation. *Journal of Information Processing*, 26, 529–538.

Dakwale, P. (2017). Fine-tuning for neural machine translation with limited degradation across in-and out-of-domain data. In *Proceedings of MT Summit XVI*.

Denkowski, M. & Lavie, A. (2010). Choosing the right evaluation for machine translation an examination of annotator and automatic metric performance on human judgment tasks. In *Proc. of AMTA 2010*.

Dorr, B., Jordan, P., & Benoit, J. (2002). A survey of current paradigms in machine translation. *Advances in Computers*, 49, 1–68.

Duh, K., Neubig, G., Sudoh, K., & Tsukada, H. (2013). Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 678–683). Sofia, Bulgaria: Association for Computational Linguistics.

Durrani, N., Sajjad, H., Joty, S., Abdelali, A., & Vogel, S. (2015). Using joint

models for domain adaptation in statistical machine translation. In *Proceedings of the 15th MT Summit (AMTA)*.

Edunov, S., Ott, M., Auli, M., & Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 489–500). Brussels, Belgium: Association for Computational Linguistics.

Escribe, M. (2019). Human evaluation of neural machine translation: The case of deep learning. In *Proceedings of the Human-Informed Translation and Interpreting Technology Workshop (HiT-IT 2019)* (pp. 36–46). Varna, Bulgaria: Incoma Ltd., Shoumen, Bulgaria.

Fernandez, J. & Downey, D. (2018). Sampling informative training data for RNN language models. In *Proceedings of ACL 2018, Student Research Workshop* (pp. 9–13). Melbourne, Australia: Association for Computational Linguistics.

Freitag, M. & Al-Onaizan, Y. (2016). Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

Gamallo, P., Pichel, J. R., & Alegria, I. (2017). A perplexity-based method for similar languages discrimination. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)* (pp. 109–114). Valencia, Spain: Association for Computational Linguistics.

Garmash, E. & Monz, C. (2016). Ensemble learning for multi-source neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 1409–1418). Osaka, Japan: The COLING 2016 Organizing Committee.

Goto, I. & Tanaka, H. (2017). Detecting untranslated content for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation* (pp. 47–55). Vancouver: Association for Computational Linguistics.

Gülçehre, Ç., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H., Bougares, F., Schwenk, H., & Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535.

Kalchbrenner, N. & Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1700–1709). Seattle, Washington, USA: Association for Computational Linguistics.

Khayrallah, H., Kumar, G., Duh, K., Post, M., & Koehn, P. (2017). Neural lattice search for domain adaptation in machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (pp. 20–25). Taipei, Taiwan: Asian Federation of Natural Language Processing.

Kobus, C., Crego, J. M., & Senellart, J. (2016). Domain control for neural machine translation. *CoRR*, abs/1612.06140.

Koehn, P. (2017). Neural machine translation. *CoRR*, abs/1709.07809.

Koehn, P. & Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation* (pp. 28–39). Vancouver: Association for Computational Linguistics.

Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 127–133).

Kudo, T. & Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Lample, G., Ott, M., Conneau, A., Denoyer, L., & Ranzato, M. (2018). Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018*

Conference on Empirical Methods in Natural Language Processing (pp. 5039–5049). Brussels, Belgium: Association for Computational Linguistics.

Ling, W., Trancoso, I., Dyer, C., & Black, A. W. (2015). Character-based neural machine translation. *CoRR*, abs/1511.04586.

Liu, N., Liu, Y., Logan, B., Xu, Z., Tang, J., & Wang, Y. (2018). Deep reinforcement learning for dynamic treatment regimes on medical registry data. *CoRR*, abs/1801.09271.

Long, Z., Kimura, R., Iida, S., Utsuro, T., Mitsuhashi, T., & Yamamoto, M. (2019). Hybrid mt based on nmt model trained with phrase tokens and large vocabulary phrase translation by smt (日本語論文:フレーズ・トークン込みnmtモデルおよびsmtによる大語彙フレーズ翻訳によるハイブリッド翻訳方式). *IEICE(Institute of Electronics, Information and Communication Engineers) Transactions*.

Luo, W. & Hayashi, Y. (2015). Utilizing alignment to compute cross-lingual semantic similarity (日本語論文:異言語文間の意味的類似度計算におけるアライメントの利用). In *Proceedings of the 2015 Conference on Association for Natural Language Processing* (pp. 63–66).

Luong, M.-T. & Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 2015 International Workshop on Spoken Language Translation (IWSLT)*.

Luong, T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1412–1421). Lisbon, Portugal: Association for Computational Linguistics.

Matusov, E., Ueffing, N., & Ney, H. (2006). Computing consensus translation for multiple machine translation systems using enhanced hypothesis alignment. In

11th Conference of the European Chapter of the Association for Computational Linguistics Trento, Italy: Association for Computational Linguistics.

Miceli Barone, A. V., Haddow, B., Germann, U., & Sennrich, R. (2017). Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 1489–1494). Copenhagen, Denmark: Association for Computational Linguistics.

Nagao, M. (1981). A framework of a mechanical translation between japanese and english by analogy principle. *International NATO Symposium on Artificial & Human Intelligence, 1981*.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (pp. 160–167). Sapporo, Japan: Association for Computational Linguistics.

Och, F. J. & Ney, H. (2001). Statistical multi-source translation. In *MT Summit 2001* (pp. 253–258).

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 311–318). Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Park, J., Song, J., & Yoon, S. (2017). Building a neural machine translation system using only synthetic parallel data. *CoRR*, abs/1704.00253.

Poncelas, A., Shterionov, D., Way, A., de Buy Wenniger, G. M., & Passban, P. (2018). Investigating backtranslation in neural machine translation. *CoRR*, abs/1804.06189.

Popovic, M. (2011). Hjerson: An open source tool for automatic error classification of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 96, 59–68.

Rikters, M. (2016). Neural network language models for candidate scoring in hybrid multi-system machine translation. In *Proceedings of the Sixth Workshop on Hybrid Approaches to Translation (HyTra6)* (pp. 8–15). Osaka, Japan: The COLING 2016 Organizing Committee.

Rokach, L. (2009). Ensemble-based classifiers. *Artificial Intelligence Review*, 33, 1–39.

Sajjad, H., Durrani, N., Dalvi, F., Belinkov, Y., & Vogel, S. (2017). Neural machine translation training in a multi-domain scenario. *CoRR*, abs/1708.08712.

Schroeder, J., Cohn, T., & Koehn, P. (2009). Word lattices for multi-source translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)* (pp. 719–727). Athens, Greece: Association for Computational Linguistics.

Schwartz, L. (2008). Multi-source translation methods. In *The 8th Biennial Conference of the Association for Machine Translation*.

Sennrich, R. (2012). Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 539–549). Avignon, France: Association for Computational Linguistics.

Sennrich, R., Haddow, B., & Birch, A. (2015). Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

Sennrich, R., Haddow, B., & Birch, A. (2016a). Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies (pp. 35–40). San Diego, California: Association for Computational Linguistics.

Sennrich, R., Haddow, B., & Birch, A. (2016b). Edinburgh neural machine translation systems for WMT 16. *CoRR*, abs/1606.02891.

Sennrich, R., Haddow, B., & Birch, A. (2016c). 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)* (pp. 1715–1725). Berlin, Germany: Association for Computational Linguistics.

Sennrich, R., Schwenk, H., & Aransa, W. (2013). A multi-domain translation model framework for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 832–840). Sofia, Bulgaria: Association for Computational Linguistics.

Servan, C., Crego, J. M., & Senellart, J. (2016). Domain specialization: a post-training domain adaptation for neural machine translation. *CoRR*, abs/1612.06141.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Temnikova, I. (2010). Cognitive evaluation approach for a controlled language post-editing experiment. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)* Valletta, Malta: European Language Resources Association (ELRA).

Van der Wees, M., Bisazza, A., & Monz, C. (2017). Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 1400–1410). Copenhagen, Denmark: Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Vauquois, B. & Boitet, C. (1985). Automated translation at grenoble university. *Computational Linguistics*, 11(1), 28–36.

Vilar, D., Xu, J., D'Haro, L. F., & Ney, H. (2006). Error analysis of statistical machine translation output. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)* Genoa, Italy: European Language Resources Association (ELRA).

Wang, R., Finch, A., Utiyama, M., & Sumita, E. (2017a). Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 560–566). Vancouver, Canada: Association for Computational Linguistics.

Wang, R., Utiyama, M., Finch, A., Liu, L., Chen, K., & Sumita, E. (2018). Sentence selection and weighting for neural machine translation domain adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, PP, 1–1.

Wang, R., Utiyama, M., Liu, L., Chen, K., & Sumita, E. (2017b). Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 1482–1488). Copenhagen, Denmark: Association for Computational Linguistics.

Wang, R., Zhao, H., Lu, B.-L., Utiyama, M., & Sumita, E. (2016). Connecting phrase based statistical machine translation adaptation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 3135–3145). Osaka, Japan: The COLING 2016 Organizing Committee.

Xia, Y., He, D., Qin, T., Wang, L., Yu, N., Liu, T., & Ma, W. (2016). Dual learning for machine translation. *CoRR*, abs/1611.00179.

Zhang, J. & Zong, C. (2016a). Bridging neural machine translation and bilingual dictionaries. *CoRR*, abs/1610.07272.

Zhang, J. & Zong, C. (2016b). Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1535–1545). Austin, Texas: Association for Computational Linguistics.

Zoph, B., Yuret, D., May, J., & Knight, K. (2016). Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1568–1575). Austin, Texas: Association for Computational Linguistics.

Appendix

The following list presents the sources (web links) for the example Japanese sentences. [1]

- p. 57 「SSML タグを使用して韻律を修正する」
  https://developer.amazon.com/ja-JP/docs/alexa/custom-skills/voice-design-best-practices-legacy.html
- p. 58 「課金管理者および課金権限について詳しくは、アクセス制御の概要をご覧ください」
  https://cloud.google.com/?hl=ja
- p. 58 「次の操作を実行する必要があるユーザーに、課金管理者の役割を割り当てます。」
  http://opus.nlpl.eu/
- pp. 58, 69 「タッチパッドはタイピング時のホームポジションをできるだけ崩さないように、パームレストの中央、もしくはそのやや左側に配置されている」
  https://ja.wikipedia.org/wiki/タッチパッド
- p. 67 「マッカーシーの 1958 年の論文では、2 つのタイプの表現が導入されている」
  https://ja.wikipedia.org/wiki/LISP
- pp. 67, 108 「ログオン認証パケット の不達や遅延によるアクセス不能問題を起こしやすい」
  https://ja.wikipedia.org/wiki/Active_Directory

[1] Content of the web link may have been edited or removed by the site administrator.

- p. 86 「労働部長官は、この日決まった同案に対する異議の申し立て期間を経て、8 月 5 日までに来年度の最低賃金を確定・告示する。」
  http://www.donga.com/jp/article/all/20090701/308222/1/
- p. 87 「名高き日本の海に浮かぶ神社」
  https://ja.japantravel.com/広島/宮島-厳島神社/9547
- p. 87 「問題:Autodesk®Fabrication データベースからデータをエクスポートし、別のいずれかにデータをインポートする必要があります。」
  https://knowledge.autodesk.com/ja
- p. 110「同一のグループプロパティ名を持つマテリアル内の全パラメータは、インスタンスのカテゴリ配下にリストされます」
  https://docs.unrealengine.com/ja/Engine/Rendering/Materials/ExpressionReference/Textures/index.html
- p. 111 「エアコン、専用バスルーム、液晶衛星テレビ、無料 Wi-Fi が備わる客室です。」
  https://ja.ehotelsreviews.com/ho299753/hotel-terminal/ja/
- p. 112 「問題:複製されたマーク値を削除しますか?」
  http://opus.nlpl.eu/
- p. 113 「ギリシャなど欧州に端を発した財政危機の波紋が広がり、我が政府や政界も財政健全性管理に向けた対策作りに苦心している。」
  http://www.donga.com/jp/article/all/20100212/310373/1/
- p. 114 「協会は、新たに八百長容疑が明らかになった 14 人の力士に対しても、独自調査を進めている。」
  http://www.donga.com/jp/article/all/20110207/412364/1/